Multimodal Human Action Recognition Based on a Fusion of Dynamic Images using CNN descriptors

Edwin Escobedo Cardenas, Guillermo Camara Chavez
Department of Computer Science (DECOM)
Federal University of Ouro Preto (UFOP)
Ouro Preto, MG, Brazil
Email: edu.escobedo88, gcamarac@gmail.com

***Abstract*— In this paper, we propose the use of dynamic-images-based approach for action recognition. Specifically, we exploit the multimodal information recorded by a Kinect sensor (RGB-D and skeleton joint data). We combine several ideas from rank pooling and skeleton optical spectra to generate dynamic images to summarize an action sequence into single flow images. We group our dynamic images into five groups: a dynamic color group (DC); a dynamic depth group (DD) and three dynamic skeleton groups (DXY, DYZ, DXZ). As action is composed of different postures along time, we generated N different dynamic images with the main postures for each dynamic group. Next, we applied a pre-trained flow-CNN to extract spatiotemporal features with a max-mean aggregation. The proposed method was evaluated on a public benchmark dataset, the UTD-MHAD, and achieved the state-of-the-art result.**

I. Introduction

Human action recognition is one of the leading components in the recent research field of human-computer interaction (*HCI*) and one of the most important topics in computer vision. It can be used as a natural and welcoming interface of interaction by users in *HCI* systems. These systems use movement and pose patterns to identify, learn and generalize actions executed by a user. Due to the enormous possibilities for practical application, there are several applications in the area of action recognition [1]: video surveillance; robotics; games; among others.

The downside of video-based methods for action recognition are the intensity images, which are vulnerable to illumination variations and cluttered backgrounds hindering the body detection and tracking. However, with the development and emergence of depth sensors, such as Microsoft Kinect [2], human action recognition from RGB-D data has attracted attention from several researchers [2]–[4]. Moreover, the Kinect sensor allows the acquisition of 3D data, can be used to capture body movements and offers 3D coordinates for the joints (skeleton data). This skeleton data is commonly available as input for human action and gesture recognition [5]–[7].

In the last years, several approaches have been proposed for human action recognition in the literature [8]–[11], *e.g.*, Chen et al. [12] proposed a method that merges the probability outputs of depth features from Kinect and inertial signal features from inertial sensor to feed collaborative representative classifiers. Imran et al. [13] proposed a deep convolutional neural network to classify human actions based on RGB-D data. First, the authors generated Motion History Images (MHIs) from RGB videos and three Depth Motion Maps (DMMs) from depth data corresponding to the front, side and top views. Zhang et al. [14] presented a feature descriptor and a decision-level fusion method for action recognition, called 3D histograms of texture (3DHoTs), that combines depth maps and texture description from a depth video sequence. They modified the Adaboost optimization function by adding the inequality constraints from SVMs in the decision-level fusion. In [15], the authors developed an integrated system that supports natural human-computer interaction and primitive cognitive task, called Cognitive Immersive Room (CIR). This system combines multimodal modalities (action, identity, attention and speech transcription) to understand or disambiguate the user intention. First, they evaluated the core techniques such as gesture and face recognition, and head pose estimation. Then, they evaluated the system by several use cases as language learning, meeting assistance, and user registration.

Currently, there are mainly two ways of using deep learning techniques to capture the spatiotemporal information in video sequences [16]: Recurrent Neural Networks (RNNs) and Convolutional Neural Networks(CNNs). RNNs are used to capture temporal information from extracted spatial skeleton features. In contrast, CNNs directly extract information from texture images which are encoded in skeleton sequences [17], [18] or from RGB-D data. Some recent works focus on the second way and propose different approaches to generate texture images for extracting relevant spatiotemporal features. Hou et al. [17] adopted Skeleton Optical Spectra (SOS) to encode dynamic spatial-temporal information. Wang et al. [18] used Joint Trajectory Maps to encode joint trajectories of the body (positions, motion directions, and motion magnitudes) of each time instance into HSV images. Li et al. [19] adopted joint distances as spatial features and a color bar for color encoding. Ding et al. [16] proposed an approach for encoding five spatial skeleton features into images with different encoding methods. Other authors followed the same ideas presented previously and proposed methods to encode video sequences into movement maps. In [20]–[22], the authors proposed a process named rank pooling to generate a unique dynamic image that summarizes an action sequence with posture and motion information to be processed by a CNN architecture.

Based on these ideas, in this paper, we propose a multimodal

human action recognition method that exploits the RGB-D and skeleton joint data recorded by a Kinect sensor. We focus on generating movement maps or dynamic images to encode spatiotemporal information. For skeleton joint data, we extend the method proposed in [17] to generate different spectra channels for each body part. For RGB-D data, we use the rank pooling process to generate dynamic color images and dynamic depth images. Moreover, we propose generating N different dynamic images to represent the main postures in an action sequence. To extract spatiotemporal features from the dynamic images, we use a pre-trained flow-CNN with a max-mean aggregation process.

The remainder of this paper is organized as follows. In Sec. II, we describe our proposed approach. Experimental results are presented in Sec. III. In Sec. IV, we discuss the conclusion and future works.

II. Method Overview

Our approach focuses on processing multimodal information (RGB-D and skeleton joint positions) recorded by a Microsoft Kinect sensor. As shown in Fig. 1, the proposed method consists of three main components: dynamic images generation, feature extraction with max-mean aggregation and classification.

A. Dynamic Images Generation

We combine several ideas from rank pooling and skeleton optical spectral to generate Dynamic Images (DI). Each one is discussed in turn. Our goal is to summarize efficiently an action of the video sequence in single flow images with posture and motion information, which can be processed later by a standard CNN architecture.

1) Skeleton Optical Spectra: Several approaches proposed a map generation from images to represent the skeleton joint data [16]–[19]. Inspired by [17], the color texture images (named Skeleton Optical Spectra or SOS images) are used to encode the skeleton joint data to capture spatiotemporal features.

Mapping of Joint Distance: Let $p_j = (p_x, p_y, p_z)$ be the coordinates of the *jth* joint in each frame, where $j \in \{1, \ldots, m\}$ and *m* is the number of joints. For each subject, the *m* joints (skeleton) from all subjects in each frame can be represented as: $s = p_1, p_2, \ldots, p_m$ and the numbering of joints follows a fixed order to maintain the correspondence between frames. Thus, a skeleton sequence of an action *A* is expressed as follows:

$$A = \{s^1, s^2, \ldots, s^n\} \tag{1}$$

where $s^i = p_1^i, p_2^i, \ldots, p_m^i$ indicates the ith skeleton of A and p_j^i represents the 3D coordinates of the jth joint in s^i.

Unlike [17], we first convert each p_j^i joint to a new origin of coordinates to avoid the translation problems of the user position regarding the Kinect. Therefore, our new joint coordinates are defined as follows:

$$p_j^{i_n} = (p_j^i - p_{new}) \tag{2}$$

In this work, we consider the shoulder center position as the new origin called p_{new}. For an action video, the skeleton joints are projected on three orthogonal Cartesian planes: *XY, YZ*, and *XZ*. Hence, we generate three sparse scatter plots in each Cartesian plane for generating the dynamic SOS images *DXY, DYZ*, and *DXZ*.

Spectrum Coding, Joint Velocity Weighted Saturation, and Brightness: Similar to [17], we use the HSB color model to generate our SOS images. To further enhance the encoded spatiotemporal information, we encode the velocity of the joints into the saturation (S) and brightness (B) of the SOS images. Moreover, we consider the relevant movement regarding each body part from the skeleton to distinguish each one. Arms and legs often have more motion information, but different frequency. Therefore, it is not recommend grouping these body parts in the same spectra. Thus, we generate five spectral distributions (H) to encode five different body parts with its respective joints: left leg part K_1 = {*left hip, left knee, left ankle, left foot*}, right leg part K_2 = {*right hip, right knee, right ankle, right foot*}, left arm part K_3 = {*left shoulder, left elbow, left wrist, left hand*}, right arm part K_4 = {*right shoulder, right elbow, right wrist, right hand*}, and middle body part K_5 = {*head, neck, torso, hip center*}.

The spectrum, *i.e.* the range of hue (H) in Eq. 3, of the right arm part, is the reversed spectrum assigned to the left arm part. The range of hue of the right leg part is the reversed spectrum assigned to the left leg part. For the middle body part, we adopt a gray scale from light gray to black (we assign $hue = 0$), because of the subtle motion of these joints [17].

In general, the encoding and enhancement of hue (H), saturation (S), and brightness (B) can be expressed as follows:

$$\begin{aligned}
H(j,i) &= \begin{cases}
\frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2} + \frac{h_{\min}}{2}, & j \in K_1 \\
\frac{h_{\max}}{2} - \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_2 \\
h_{\max} - \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_3 \\
\frac{h_{\max}}{2} + \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_4 \\
0, & j \in K_5
\end{cases} \\
S(j,i) &= \begin{cases}
\frac{v_j^i}{\max\{v\}} \times (s_{\max} - s_{\min}) + s_{\min}, & j \in K_{1:4} \\
0, & j \in K_5
\end{cases} \\
B(j,i) &= \begin{cases}
\frac{v_j^i}{\max\{v\}} \times (b_{\max} - b_{\min}) + b_{\min}, & j \in K_{1:4} \\
b_{\max} - \frac{i}{n} \times (b_{\max} - b_{\min}), & j \in K_5
\end{cases}
\end{aligned} \tag{3}$$

where the joint velocity is calculated by

$$v_j^i = \left\| p_j^{i+1} - p_j^i \right\|_2. \tag{4}$$

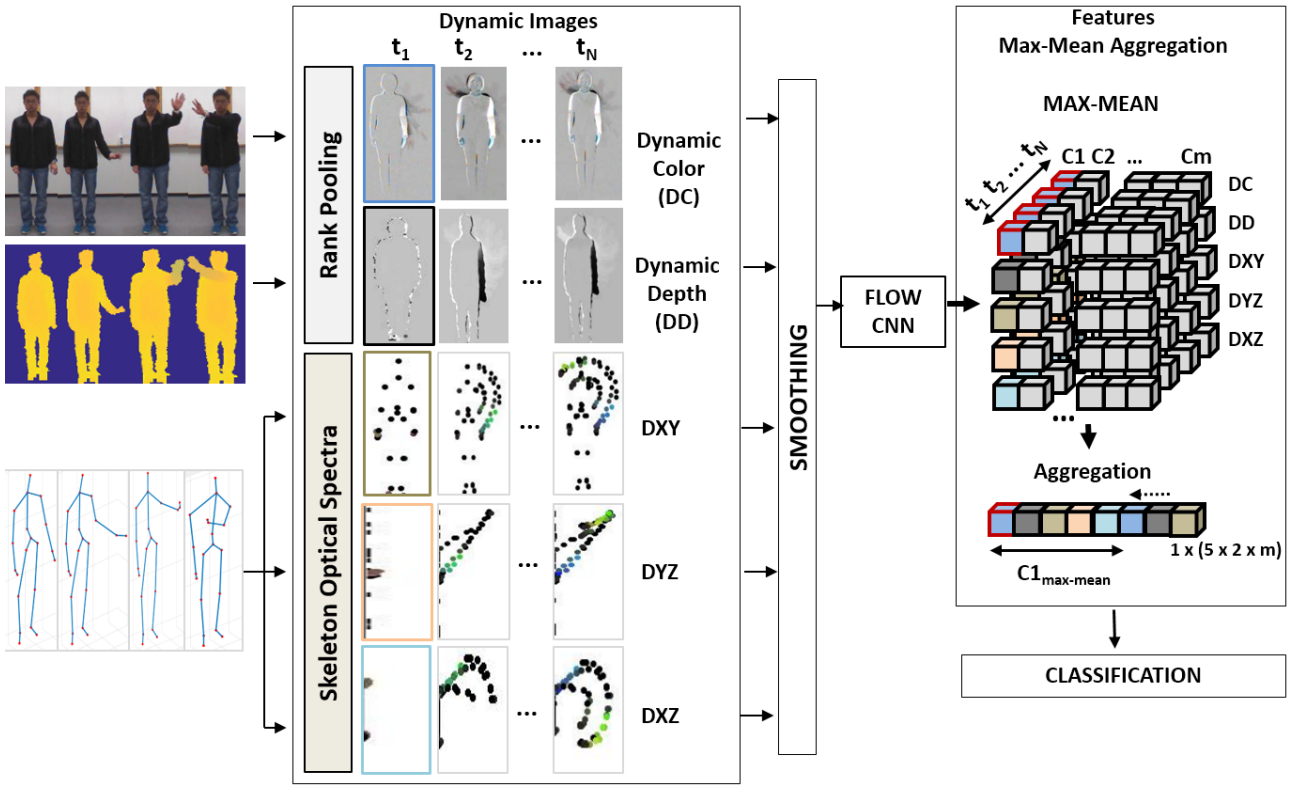


Fig. 1. Overview of our proposed method for action recognition. Multimodal data is used to extract spatiotemporal features. We generate Dynamic Images (*DI*) to summarize the motion and posture from an action divided in N main postures for a particular time t_i. Next, we smooth the t_N dynamic images to improve the image quality and highlight regions with movement. Then, we apply a pre-trained flow-CNN on the *DIs* to extract our features. Finally, we apply a max-mean aggregation process to integrate all features and use a linear SVM classifier with the goal to boost the recognition performance.

(a) DXY (b) DYZ (c) DXZ

Fig. 2. Dynamic SOS images generated for each Cartesian plane (using the skeleton joint data).

Thus, for each joint p_j^i in the skeleton s^i of an action A, we apply the Eq. 3 to compute its respective hue, saturation and brightness values. Next, these values are plotted in each Cartesian plane to generate three Dynamic SOS images (*DXY, DYZ, DXZ*). Finally, we convert the SOS images to RGB color model to process them with a pre-trained flow-CNN network. Fig. 2 shows an example of the *DXY, DYZ, DXZ* dynamic SOS images generated for a particular action.

2) Rank Pooling: For RGB-D data, we use the dynamic images generation based on rank pooling proposed in [21], [22]. The core idea is to represent a video through a standard RGB image that summarizes the appearance and dynamics of a whole video sequence.

We can represent an action video as a ranking function for its frames $I_1, \ldots, I_T$. In more detail, let $\psi_{I_t} \in \mathrm{R}^d$ be a representation or feature vector extracted from each individual frame I_t in the video. Let $V_i = \frac{1}{i}\sum_{\tau=1}^{i} \psi(I_\tau)$ be time average of these features up to time t [22]. The ranking function associates each time t a score $S(t|\mathbf{d}) = \langle \mathbf{d}, V_t \rangle$, where $\mathbf{d} \in \mathbb{R}^d$ is a vector of parameters. The function parameters $\mathbf{d}$ are learned, so that the scores reflect the rank of the frames in the video. Therefore, later times are associated with larger scores, *i.e.* $q > t \implies S(q|\mathbf{d}) > S(t|\mathbf{d})$. Learning $\mathbf{d}$ is posed as a convex optimization problem using the *RankSVM* [23] formulation:

$$\mathbf{d}^* = \rho(I_1, \ldots, I_T; \psi) = \arg\min_{\mathbf{d}} \quad E(\mathbf{d}) \tag{5}$$

$$E(\mathbf{d}) = \frac{\lambda}{2}\|\mathbf{d}\|^2 + \frac{2}{T(T-1)} \times \sum_{q>t} \max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\}. \tag{6}$$

Thus, computing a dynamic image entails solving the optimization problem of Eq. 5. In [21], the authors presented an approximation to rank pooling which is much faster and works well in practice. They derivated the approximate rank

(a) Dynamic Color (DC)

(b) Dynamic Depth (DD)

Fig. 3. Dynamic Images generated using ranking pooling on the RGB-D data.

pooling based on the idea of considering the first step in a gradient-based optimization reducing the Eq. 5 to:

$$\hat{\rho}(I_1, \ \ldots, \ I_T; \psi) = \sum_{t=1}^{T} \alpha_t \psi(I_t). \tag{7}$$

The coefficients α_t are given by:

$$\alpha_t = 2(T - t + 1) - (T + 1)(H_T - H_{t-1}). \tag{8}$$

where $H_t = \sum_{i=1}^{t} 1/t$ is the $t - th$ Harmonic number and $H_0 = 0$. Likewise, we can use directly individual video frames I_t replacing $\psi(I_t)$.

Hence, the authors show that d^* can be interpreted as a standard RGB image. Furthermore, since this image is obtained by rank pooling the video frames, it summarizes information from the whole video sequence. The complete process is explained and detailed in [21], [22].

To generate dynamic images from depth data, we normalized each video frame v_d to the interval $[0 : 255]$ using the Min-Max normalization defined by:

$$v_d = \frac{v_d - min(v_d)}{max(v_d) - min(v_d)} \times 255 \tag{9}$$

Finally, we generate two dynamic images from RGB-D videos as shown in Fig. 3. In Fig. 3a, we show the dynamic color image (DC) computed for all frames from RGB video. In Fig. 3b, is presented the dynamic depth image (DD) computed from the normalized depth video. It is possible to notice that dynamic images tend to focus mainly on the active body part, such as the right arm in Fig. 3. In contrast, background pixels and background motion patterns tend to be averaged. Hence, the pixels in dynamic images seem to focus on the appearance and motion of the user body, which indicates that they can contain the necessary information to recognize the action.

B. Multiple Dynamic Images Generation

As we can see, the generation of a dynamic image for each multimodal channel summarizes an action sequence more efficiently. Nevertheless, different approaches showed that is possible to divide an action into different relevant sequences [5], [24]. In consequence, we can represent an action by different postures $A = \{A_1, A_2, \ldots, A_T\}$ along time T. It is possible to identify the N main postures which are more relevant in A, so we generate a dynamic image for each one. In this manner, we create N dynamic images for an action video dividing it into N sub-movements from the time $t_{ini} = 1$ to t_i, where $t_i = \{t_1, t_2, \ldots, t_N\}$ and t_N represent the complete dynamic image along time $[1 : T]$. In Fig. 4, we show *DIs* generated for an action divided into $N = 3$ sub-movements.

C. Feature Extraction and Aggregation

To improve the dynamic image quality (contrast enhancement) and highlight regions with movement, we first apply a smoothing process by an isotropic Gaussian kernel with a standard deviation $\sigma = 3$ for all the generated dynamic images. Next, based on the transfer learning property of the CNNs [25] and the ideas presented in [26], we use a pre-trained flow-CNN architecture proposed in [27] to process the dynamic images. The flow-CNN network contains five convolutional and three fully-connected layers. It was trained with optical flow images to compute robust descriptors from dynamic images. The output of the second fully-connected layer with $m = 4096$ values is used as our feature vector. Therefore, for each dynamic image group DI^j (DC, DD, DXY, DYZ, DXZ) we computed N feature vectors $f^j_{t_i}$, where $t_i \in \{t_1, t_2, \ldots, t_N\}$, and m are the dimension of $f^j_{t_i}$. Finally, we obtained a three-dimensional matrix of spatiotemporal features M_{jmN}, as shown in Fig. 1.

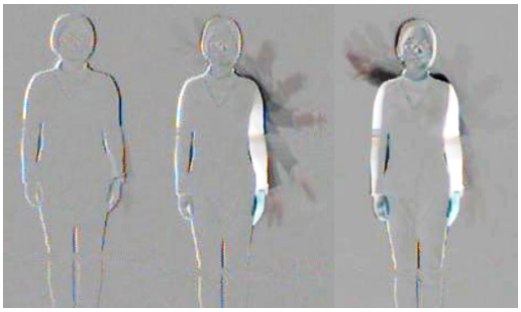
To integrate the spatiotemporal features presented in the M_{jmN} matrix, we follow these simple steps: First, we apply a max-mean function between the t_N dynamic images for a DI^j group, *i. e.* we apply the max-mean operators between the M_m and M_N dimensions, so we obtain the most discriminative features for each DI^j group. Next, we apply an aggregation process to convert the M_{j2m} matrix to one-dimensional feature vector $F_{1\times(10m)}$.

III. Experimental Results

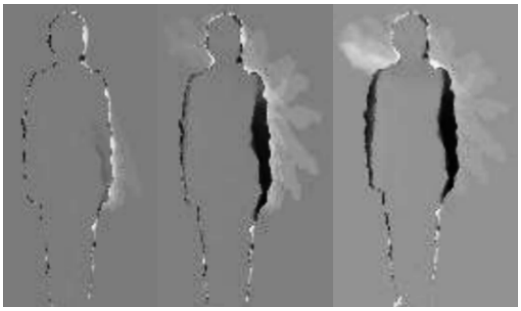
In this section, we present the parameter setting for our experiments. Then, we evaluate the performance of our proposed method by testing the discriminative ability of our spatiotemporal features on a public action dataset.

A. Dataset

The UTD-MHAD dataset was collected using Kinect V1 sensor and a wearable inertial sensor in an indoor environment [28]. The dataset contains 27 actions performed by 8 subjects (4 females and 4 males). Each subject repeated each action 4 times. The dataset includes 861 data sequences. Four data modalities of RGB videos, depth videos, skeleton joint positions, and the inertial sensor signals were recorded in three channels. One channel was used for simultaneous capture of depth videos and skeleton positions, one channel for RGB videos, and one channel for the inertial sensor signals (3-axis acceleration and 3-axis rotation signals). Fig. 5 illustrates an example of the multimodal data corresponding to the action basketball-shoot.



(a) Dynamic Color Images



(b) Dynamic Depth Images

(c) Dynamic DXY images

(d) Dynamic DYZ Images

(e) Dynamic DXZ Images

Fig. 4. Dynamic Images generated for $N = 3$ in each multimodal channel.

To conduct experiments, the authors divided the dataset into two sub-datasets for training and testing that are mutually exclusive. Moreover, we only use the Kinect data without including the inertial information. Table I shows the detailed information of the UTD-MHAD dataset.

TABLE I
EXPERIMENTAL INFORMATION FOR THE UTD-MHAD DATASET.

Sets	Gestures	RGB	Depth	Skeleton	Subjects
Training	432	432	432	432	4 (1,3,5,7)
Testing	432	432	432	432	4 (2,4,6,8)

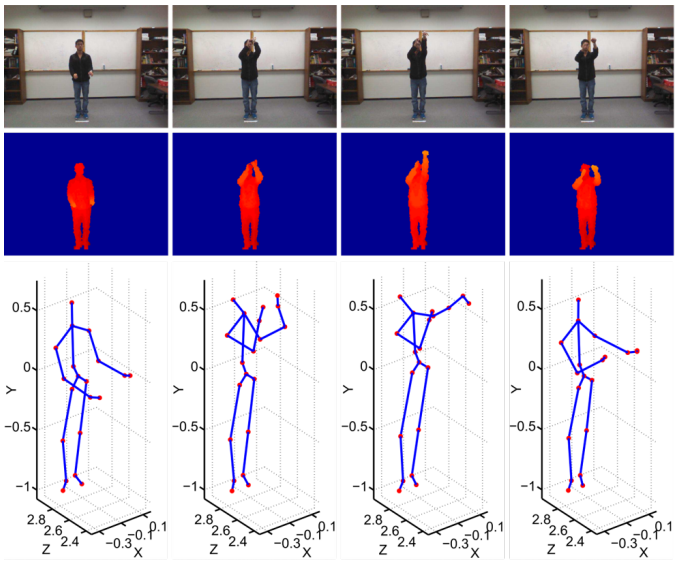


Fig. 5. An example of the multimodal data corresponding to the action basketball-shoot in the UTD-MHAD dataset [28]. The first row shows the color images; the second row, the depth images (the background of each depth frame was removed); the third row, the skeleton joint frames.

B. Parameter setting

1) All experiments were conducted and measured in a notebook with a CPU *Intel Core i7 inside, 2.5 GHz U*, *12* GB of memory and a *GPU GEFORCE GTX 950M* with 4GB of memory.
2) The proposed method was implemented using the MATLAB development IDE R2016a (64 bits).
3) In all experiments, we used Support Vector Machines (*SVM*) [29] with linear kernel using the LIBSVM library [30].
4) Results were obtained from the testing dataset and reported on all Tables. Additionally, we compared our method with the state-of-the-art using the experimental protocol indicated by the authors.
5) In all experiments, we consider: $h_{min} = 0$, $h_{max} = 360$, $s_{min} = 0$, $s_{max} = 1.0$, $b_{min} = 0$, and $b_{max} = 1.0$.

C. Finding the Optimal value for N postures

We conducted an initial experiment using the training dataset to find the optimal value of N to compute the t_N dynamic images for each DI^j group. We used only dynamic skeleton images as input because it is most feasible to observe

results in the DXY generated images. As the UTD-MHAD dataset does not provide a validation set, we used cross-validation with $K = 4$ folds.

We tested six N_i possible values $N_i = \{1, 2, 3, 4, 5, 6\}$. Results are shown in Fig. 6. We obtained top results for $N \leq 4$. When N is bigger (i. e., $N > 4$), the recognition rate started to drop greatly owing to the inconsistency in the skeleton joint distribution that assigns a similar quantity of skeleton points for a t_i posture. This problem generates dynamic images with poor dissimilarity and redundant information (Fig. 7). Consequently, we used the best value $N = 4$ to conduct the rest of the experiments.

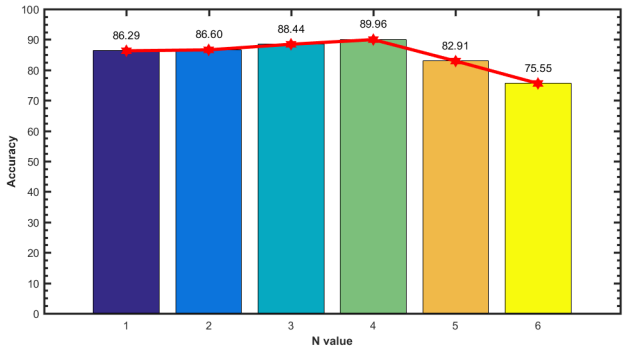


Fig. 6. Results for different values of N; for $N = 1$, we generated only a dynamic image using all skeleton joint positions.

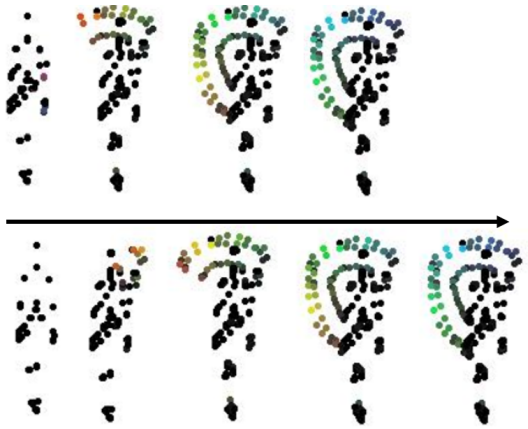


Fig. 7. Dynamic DXY images for $N = 4$ (up) and $N = 5$ (bottom). When $N = 4$, we obtained dynamic images with a major dissimilarity. For $N = 5$, there is an inconsistency in the skeleton joint distribution that generates dynamic images with poor information. This inconsistent increases when N is bigger, generating a loss in the recognition rate.

D. Evaluation of Different Encoding Schemes

We defined and evaluated the effectiveness of different encoding schemes to determine which method capture spatiotemporal information effectively. Following, we list the schemes to be evaluated:

- **SCH01** (DC_{N1}): uses only dynamic color images with $N = 1$ postures (a unique dynamic image for an action video) in the classification process.
- **SCH02** (DD_{N1}): uses only dynamic depth images with $N = 1$ postures in the classification process.
- **SCH03** $(DXY+DYZ+DXZ)_{N=1}$: uses only dynamic skeleton images with $N = 1$ postures in the classification process.
- **SCH04** $(DC + DD + DXY + DYZ + DXZ)_{N=1}$: combines dynamic images from color, depth and skeleton with $N = 1$ postures in the classification process.
- **SCH05** (DC_{N4}): uses only dynamic color images with $N = 4$ postures in the classification process.
- **SCH06** (DD_{N4}): uses only dynamic depth images with $N = 4$ postures in the classification process.
- **SCH07** $(DXY+DYZ+DXZ)_{N=4}$: uses only dynamic skeleton images with $N = 4$ postures in the classification process.
- **SCH08** $(DC + DD + DXY + DYZ + DXZ)_{N=4}$: combines dynamic images from color, depth and skeleton with $N = 4$ postures for each dynamic image type in the classification process.

1) Unique Dynamic Image vs. Dynamic Images with different posture: From Table II, we can see that the integration of all dynamic image features effectively capture spatiotemporal information. Likewise, we can see that the generation of N postures for each DI^j along different times t_i, contribute to improve the recognition rate since the schemes with $N = 4$ overcome schemes with $N = 1$. A particular case, we note that **SCH03** scheme outperforms the research proposed in [17] (Table III), in which only works with three body parts (in our case we use five); this may be due to the fact that arms and legs have movement with different intensity and we grouped them in different spectrum channels. The **SCH08** scheme integrates all dynamic image features with N postures through a max-mean aggregation process and reaches the best result (94.57%). In general, the creation of N dynamic images for a particular posture provides additional spatiotemporal features that improve the action recognition rate.

The confusion matrix is shown in Fig. 8. The individual results for each action in the UTD-MHAD dataset are presented in Fig. 9. The UTD-MHAD is a challenging dataset, so we had to use a method to integrate all our dynamic image features to get the better result. However, in the confusion matrix we can see that our final **SCH08** scheme does not distinguish some actions very well, e.g. *jogging in place-22* and *walking-23* or *wave-03* and *throw-05*. There are many probable reasons for these, such as the filters of the pre-trained flow-CNN may not be adequate for distinguishing similar actions or the max-mean operators of the aggregation process which may eliminate some valuable information. In fact, we will review all these problems in future works. Finally, we take into account the use of depth data to extract relevant features. Many recent works focus on exploiting this information to obtain relevant features capable of overcoming RGB features. In our experiments,

dynamic depth images (**SCH02, SCH06**) achieve better results than dynamic color images (**SCH01, SCH05**). These results show the importance of the depth information to generate spatiotemporal features for action recognition.

TABLE II
RESULTS OF DIFFERENT ENCODING SCHEMES ON THE UTD-MHAD DATASET.

Code	Encoding Scheme	Accuracy (%)
SCH01	DC_{N1}	58.29
SCH02	DD_{N1}	78.22
SCH03	$(DXY + DYZ + DXZ)_{N=1}$	87.42
SCH04	$(DC + DD + DXY + DYZ + DXZ)_{N=1}$	**91.27**
SCH05	DC_{N4}	69.86
SCH06	DD_{N4}	86.20
SCH07	$(DXY + DYZ + DXZ)_{N=4}$	90.42
SCH08	$(DC + DD + DXY + DYZ + DXZ)_{N=4}$	**94.57**

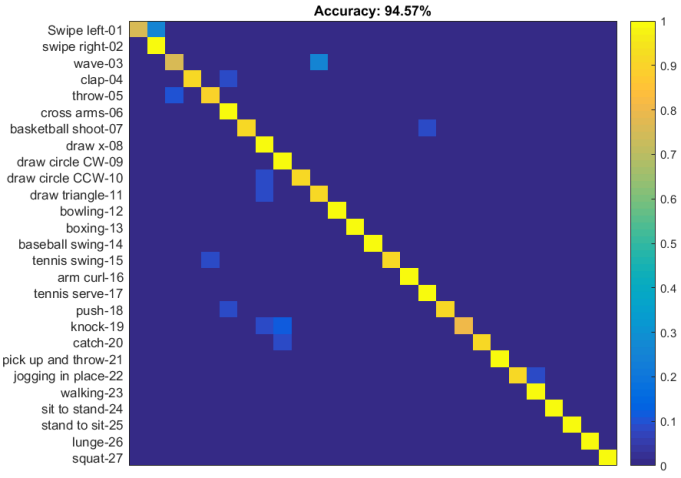


Fig. 8. Confusion Matrix from the best encoding scheme: **SCH08** $(DC + DD + DXY + DYZ + DXZ)_{N=4}$.

E. Comparison with the state-of-the-art

Table III compare the performance of the **SCH08** scheme with the state-of-the-art methods. Please notice that the method proposed by Chen et al. [12] achieved 97.2% of accuracy when used both Kinect and inertial sensor data; 85.10%, when used only Kinect data. In contrast, we achieve 94.57% of accuracy using Kinect data and overcome them under the same conditions. Other methods achieved results between 81% and 89%. Recently, methods proposed by Zhao et al. [15] (90.90%) and Imran and Kumar [13] (91.20%), achieved results greater than 90% using Kinect data only. Imran and Kumar [13] followed a line of research similar to ours. They used a deep convolutional neural network to classify human actions based on RGB-D data using Motion History Images and different modalities to fusion features. Within all methods, our **SCH08** scheme achieves the best result overcoming the second best result in 3.3% of the difference in performance.

Finally, our method recorded an average execution time of 0.26 seconds to generate only a dynamic image from a video on the UTD-MHAD dataset. To generated N dynamic images for each dynamic group ($4 \times 5 = 20$), we reached an average execution time of 4.35 seconds per video with a frame number between 45 and 125.

TABLE III
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE UTD-MHAD DATASET.

Method	Accuracy (%)
Chen and Forbus [31]	81.00
Zhang et al. [14]	84.40
Chen et al. [12] (Kinect sensor only)	85.10
Chen et al. [12] (Kinect and inertial sensors fusion)	97.20
Wang et al. [18]	85.81
Hou et al. [17]	86.97
Li and Hou [19]	88.10
Zhao et al. [15]	90.90
Imran and Kumar [13]	91.20
SCH08 $(DC + DD + DXY + DYZ + DXZ)_{N=4}$	**94.57**

IV. CONCLUSION

In this paper, we address the problem of human action recognition based on the calculation of spatiotemporal features from multimodal data recorded by a Kinect sensor (RGB-D and skeleton joint data). We combined several ideas from rank pooling and skeleton optical spectra to generate dynamic images to summarize an action sequence into single flow images. We grouped our dynamic images into five groups: a dynamic color group (DC); a dynamic depth group (DD) and three dynamic skeleton groups (DXY, DYZ, DXZ). As the action is composed of different postures along time, we generated N different dynamic images which are the main postures for each dynamic group. Next, we applied a pre-trained flow-CNN to extract spatiotemporal features with a max-mean aggregation. Experimental results showed the efficacy of the proposed method, we obtained 94.57% of accuracy and outperformed in 3.3% the second best method of the literature. Likewise, results showed that our method presented mistakes to recognize similar actions. In fact, as future work, we pretend to explore new CNN architectures and new fusion schemes to integrate the features extracted from each dynamic group to improve these limitations, so our method will be capable of differentiating similar actions.

ACKNOWLEDGMENT

The authors thank the Graduate Program in Computer Science (PPGCC) at the Federal University of Ouro Preto (UFOP), the Coordination for the Improvement of Higher Level Personneland (CAPES) and the funding Brazilian agency CNPq.

REFERENCES

[1] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[2] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

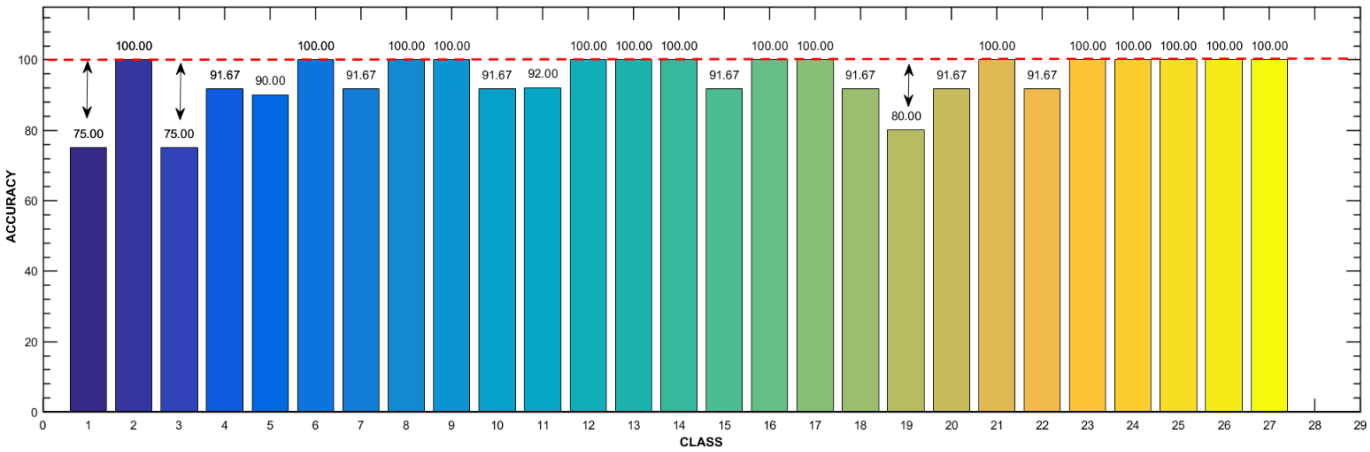


Fig. 9. Individual results on the UTD-MHAD dataset from the best encoding scheme: **SCH08** $(DC + DD + DXY + DYZ + DXZ)_{N=4}$.

[3] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[4] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang, "Multimodal multipart learning for action recognition in depth videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2123–2129, 2016.

[5] E. Escobedo and G. Camara, "A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes," in *Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference on*. IEEE, 2016, pp. 209–216.

[6] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE, 2013, pp. 471–478.

[7] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 4513–4518.

[8] G. Guo and A. Lai, "A survey on still image based human action recognition," *Pattern Recognition*, vol. 47, no. 10, pp. 3343–3361, 2014.

[9] F. Zhu, L. Shao, J. Xie, and Y. Fang, "From handcrafted to learned representations for human action recognition: A survey," *Image and Vision Computing*, vol. 55, pp. 42–52, 2016.

[10] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4–21, 2017.

[11] M. Asadi-Aghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera, "A survey on deep learning based approaches for action and gesture recognition in image sequences," in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*. IEEE, 2017, pp. 476–483.

[12] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, 2016.

[13] J. Imran and P. Kumar, "Human action recognition using rgb-d sensor and deep convolutional neural networks," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*. IEEE, 2016, pp. 144–148.

[14] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, "Action recognition using 3d histograms of texture and a multi-class boosting classifier," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4648–4660, 2017.

[15] R. Zhao, K. Wang, R. Divekar, R. Rouhani, H. Su, and Q. Ji, "An immersive system with multi-modal human-computer interaction," in *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*. IEEE, 2018, pp. 517–524.

[16] Z. Ding, P. Wang, P. O. Ogunbona, and W. Li, "Investigation of different skeleton features for cnn-based 3d action recognition," in *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*. IEEE, 2017, pp. 617–622.

[17] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton optical spectra based action recognition using convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[18] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 102–106.

[19] C. Li, Y. Hou, P. Wang, and W. Li, "Joint distance maps based action recognition with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, 2017.

[20] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Rank pooling for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 773–787, 2017.

[21] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[22] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[23] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[24] B. Pathak, A. S. Jalal, S. C. Agrawal, and C. Bhatnagar, "A framework for dynamic hand gesture recognition using key frames extraction," in *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2015 Fifth National Conference on*. IEEE, 2015, pp. 1–4.

[25] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[26] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3218–3226.

[27] G. Gkioxari and J. Malik, "Finding action tubes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 759–768.

[28] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 168–172.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] 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.

[31] K. Chen and K. D. Forbus, "Action recognition from skeleton data via analogical generalization," in *Proc. 30th International Workshop on Qualitative Reasoning*, 2017.