This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology

Dynamic 3D Hand Gesture Recognition by Learning Weighted Depth Motion Maps

Reza Azad*, Maryam Asadi-Aghbolaghi*, Shohreh Kasaei *Senior Member IEEE*, and Sergio Escalera

***Abstract*—Hand gesture recognition from sequences of depth maps is a challenging computer vision task because of the low inter-class and high intra-class variability, different execution rates of each gesture, and the high articulated nature of human hand. In this paper, a *multilevel temporal sampling* (MTS) method is first proposed that is based on the motion energy of key-frames of depth sequences. As a result, long, middle, and short sequences are generated that contain the relevant gesture information. The MTS results in increasing the intra-class similarity while raising the inter-class dissimilarities. The *weighted depth motion map* (WDMM) is then proposed to extract the spatio-temporal information from generated summarized sequences by an accumulated weighted absolute difference of consecutive frames. The *histogram of gradient* (HOG) and *local binary pattern* (LBP) are exploited to extract features from WDMM. The obtained results define the current state-of-the-art on three public benchmark datasets of: MSR Gesture 3D, SKIG, and MSR Action 3D, for 3D hand gesture recognition. We also achieve competitive results on NTU action dataset.**

***Index Terms*—Hand gesture recognition, Multilevel temporal sampling, Weighted depth motion map, Spatio-temporal description, VLAD encoding.**

I. Introduction

HAND gesture recognition from sequences of depth maps is an active research area in computer vision; because of its potential applications in sign language processing [47], video surveillance [1], medical training [2], remote controlling [3], and human-environment interaction [4]. The *hand gesture recognition* (HGR) refers to classification of dynamic hand movements in action videos. Generally, HGR can be decomposed into three main steps of hand detection, feature extraction, and classification.

Early work on hand detection was based on wearable sensors; such as data gloves. Although those sensors result in accurate measurements of hand pose and location, they are expensive and invasive and also require accurate calibration. Therefore, they are inappropriate for uncontrolled application scenarios. Fortunately, recent advances in imaging devices, like Microsoft Kinect, have received great attention from researchers to reconsider the problems such as gesture recognition from depth information [5, 6].

For feature extraction, one has to deal with different camera viewpoints, variable hand sizes, finger occlusions, and also different execution rates. These may cause high intra-class variations. Furthermore, there are some gestures with different labels for which some parts of hand movements or hand poses are equivalent, showing low inter-class variations. Consequently, video representation and feature extraction are critical steps in order to compute discriminative descriptors that are robust to aforementioned challenges. In terms of depth video representation, the so called *depth motion map* (DMM) has been used by several researches as a global temporal template to codify gestures [7, 8]. Also, DMM is formed by accumulating the absolute distance of consecutive frames from different projection views. The last step includes classification of feature descriptors. For handcrafted features, machine learning tools like *support vector machine* (SVM) and *random forest* (RF) are commonly used for classification.

Many descriptors of HGR behave differently for actions performed with different speeds. This challenge causes to increase the intra-class variations and therefore decreases the recognition accuracy. In order to make the method robust against different execution rates of each gesture, the input data is augmented by the proposed MTS method. Three temporal scales of long, middle, and short duration are generated from the original data by sampling key-frames of input depth map sequences. These key-frames are selected based on their motion energy which are estimated by the absolute difference of consecutive frames.

The WDMM is proposed as a temporal weighted version of DMM to compute one single image from a sequence of depth frames by accumulating motion energy of the projected depth maps into three projective views. The temporal weights are used to distinguish the motion direction by giving more weight to recent depth frames.

In this paper, the HOG [9] and LBP [34] descriptors are employed to extract features from WDMM. HOG descriptor is able to describe the local object appearance and shape within an image. It is formed by using the distribution of intensity gradients. On the other hand, the LBP descriptor is a powerful and effective texture descriptor. In fact, to describe the local texture patterns of an image, LBP compares every gray value to the center pixel of a neighborhood and computes a binary code by setting a threshold on those comparisons.

Next, the *vector of locally aggregated descriptors* VLAD encoding process [10] is employed to transform the local features (extracted by HOG and LBP) into a fixed-size vec-

* These two authors contributed equally.

R. Azad, M. Asadi-Aghbolaghi, and S. Kasaei are with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. Phone: +98 21 6616 6646, Fax: +98 21 6601 9246 (e-mail: razad@ce.sharif.edu, masadia@ce.sharif.edu, skasaei@sharif.edu).

S. Escalera is with the University of Barcelona and Computer Vision Center, Barcelona, Spain. Phone: +34 93 4020853, Fax: +34 93 581 16 70 (email: sergio@maia.ub.es).

tor representation. Then, to reduce the dimension of feature vectors, the *principal component analysis* (PCA) is applied on the VLAD encoded descriptor. At the last stage, the *single hidden layer feed-forward neural network* (SLFN) with *extreme learning machine* (ELM) [11] is utilized to classify hand gestures. The key contributions of this work can be summarized as follows:

- A novel MTS method based on key frame extraction is introduced for generating long, middle, and short videos.
- A new weighted method is proposed to compute the depth motion map to consider the order of temporal information.
- A compact representation of video sequence is computed by utilizing the combination of VLAD encoding of HOG and LBP visual words.
- The proposed method achieves the state-of-the-art results on MSR Gesture 3D, SKIG, and MSR Action 3D datasets and outperforming deep learning results. Moreover, the result of NTU which is the largest RGB-D available dataset, is competitive to the state-of-the-art methods[1].

The rest of this paper is organized as follow. In Section II, related work are reviewed. Section III presents the proposed video representation. Experimental results are given in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

Gestures are body motions that convey meaningful information to interact with the environment. Gestures involve physical motions of fingers, hands, face, arms, and torso. In this section, we review main methods related to HGR from sequences of depth maps. Several surveys have been published for gesture recognition from depth data [12]. There are also some work that make use of both RGB and depth data [13, 14]. During the last few years, the progress of depth sensing devices, like Microsoft Kinect, have greatly promoted the research on HGR. Microsoft Kinect includes a depth camera and a *video graphics array* (VGA) camera. Both cameras produce image streams at 30 *frames per second* (fps). Depth-based gesture recognition can be categorized into three groups of hand skeleton, spatio-temporal volume of hand, and deep learning-based methods. These are discussed next.

A. Hand Skeleton

Connected joints of hand skeleton are mostly extracted from depth maps. Two kinds of geometrical features (spatial and temporal) can be extracted from the skeletal data. In each frame, the relative positions (like Euclidean distance) of hand joints (to each other or to reference points) are extracted as spatial features. Temporal features are formed by the relative position of hand joints in each frame to the same joints in other frames. These define joint trajectories.

Smedt et al. [15] used 22 hand joints returned by the Intel RealSense camera. Movement, the rotation of hand in space and also hand shape variations based on skeleton joints are encoded using a temporal pyramid. In the work of [16], Smedt et al. represented 3D Hand gestures as a set of trajectories of relevant joints of hand-parts in the Euclidean space.

Escobedo et al. [17] used the 3D trajectory of hand skeleton in spherical coordinates to extract key frames from input sequences. Lu et al. [5] proposed single-finger and double-finger features. Euclidean distances between fingertips and a reference point and also the angles between them formed the single finger features. Double finger features includes Euclidean distances and angles between adjacent fingertips.

Unfortunately, some shortcomings limit the usage of skeletal information for both gesture and action recognition. To mention some, localizing hand joints is a time-consuming task that needs high-resolution images. Moreover, estimation of hand joints is unreliable or even fails in presence of self-occlusion.

B. Spatio-Temporal Volume of Hand

Spatio-temporal volumes of hand have been used to extract spatio-temporal features from sequences of depth maps. Elmezain et al. [18] first detected and tracked hand by using RGB-D information. They then extracted location, orientation, and velocity of hand using the spatio-temporal volume of detected hand. In [19], Kurakin el al. proposed an orientation normalization method for hand gesture recognition from depth data. The depth image was rotated in such a way that the palm is approximately parallel to the image plane. The silhouette computed from volume of gestures was divided into some cells and then features were extracted from those cells.

The 3D kernel descriptor [20] utilized the unsupervised *kernel principal component analysis* (KPCA) to learn a compact descriptor from the spatio-temporal gradient of depth data. Asadi and Kasaei [21] proposed *supervised spatio-temporal kernel descriptor* (SSTKDes) to define a discriminative and compact feature representation of depth sequences.

Yang and Tian [23] proposed the *super normal vector* (SNV), in which the hypersurface normal vectors in each spatio-temporal neighborhood were clustered to form the low level poly-normal. The *histogram of oriented 4D normals* (HON4D) [22] and DMM [7, 8] have been also extracted from spatio-temporal volumes of depth sequences. The HON4D descriptor was based on the distribution of 4D normal vectors in some spatio-temporal cells of actions. In DMM, the point cloud of a human body was projected onto three orthogonal Cartesian views. Then, the global spatio-temporal activity of the entire video sequences was accumulated on those planes. Subsequently, the HOG and LBP descriptors were utilized to form the final descriptor.

Methods based on spatio-temporal hand motion are mostly the fastest ones. However, the motion-based descriptors have their own drawbacks. For instance, sometimes there is no difference between paired sequences ("*sit-down*" and "*stand-up*"). Moreover, variations in execution rates of each gesture result in sequences with different lengths, which in turn reduces the final accuracy.

C. Deep Learning-based Methods

Asadi et al. [13] divided the main deep learning-based methods into four groups of 2D models, motion-based input

[1]Our source code is available on https://github.com/rezazad68/Dynamic-3D-Action-Recognition-on-RGB-D-Videos.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology

features, 3D models, and temporal methods. In the first category, 2D *convolutional neural network* (CNN) is utilized to extract spatial features from one or more sampled frames of the whole video. The label of the gesture is calculated by score averaging of the results of sampled frames. For sign language gesture recognition, Kang et al. [24] utilized a CNN to extract features from the fully connected layer for depth maps. To consider motion information, in the second category, simple motion features (like optical flow) are first pre-computed and then are fed to the 2D models. Wu et al. [25] exploited a two-stream CNN to learn a set of training gestures. In fact, raw depth data is fed to the spatial network and optical flow is used as the input of temporal network. Wang et al. [26] employed CNN for classification of DMMs which are extracted as basic motion features from depth sequence. Asadi et al. [27] used scene flow which is the real 3D motion of objects as the input data of CNN. Wang et al. [28] also proposed to utilize the scene flow. In order to take advantage of the available model trained over ImageNet, the scene flow is first transformed to an optimal color space analogous to RGB. They then used motion maps constructed from the sequence of scene flow data as the input data of the 2D deep model.

The 3D filters in the convolutional layers are utilized in the third category which allows to capture discriminative information along both spatial and temporal dimensions at the same time. Molchanov et al. utilized 3D CNN to recognize hand gestures. In [29], the saliency video was generated from the RGB data to focus on the salient object in the video. They then used the C3D to extract spatio-temporal features from RGB-D, and saliency videos. Late fusion was then employed to combine the features extracted from these three modalities.

Finally the fourth group uses temporal processing tools (like *recurrent neural network* (RNN) with LSTM) to process input sequences. Molchanov et al. [30] exploited both 3D CNN and RNN for recognizing hand gestures. In their method, short clips of the entire video were fed to the 3D CNN and then the outputs of 3D CNN were used as the input to RNN. Neverova et al. [31] proposed a multimodal (depth, skeleton, and speech) human gesture recognition system based on RNN. Discriminative data-specific features were either manually extracted or learned from short spatio-temporal blocks. Then, RNN was employed for modeling large-scale temporal dependencies, performing data fusion, and ultimately classifying gestures.

Zhang et al. [32] utilized the combination of 3DCNN, Convolutional LSTM, and 2DCNN for action recognition from the RGB-D data. To do that, short-term spatio-temporal features were first learned by the 3DCNN and then by employing the Convolutional LSTM, long-term spatio-temporal features were learned. This combination resulted in 2D spatio-temporal feature maps. The authors lastly recognized gestures by exploiting the 2DCNN on 2D feature maps. In [33], Luo et al. proposed an unsupervised approach to extract atomic 3D motion as the features for action recognition.

Although, deep learning-based methods have improved the performance of classical handcrafted methods in many applications (like image classification), gesture and action recognition methods have not gained a high performance from deep networks. The reason behind this is that a large annotated dataset (like ImageNet) is required to learn a large number of weights, which is not currently available for gesture recognition purposes. Finetuning deep networks for gesture recognition from pre-trained models on ImageNet results in performance improvements; however, for not very large datasets, handcrafted features still outperform deep models.

In this paper, which mostly belongs to the second category, the depth sequences of gestures are considered as spatio-temporal volumes. To make the method robust to the execution rate of each gesture, the MTS is introduced. It generates an extended set of videos with different lengths based on the motion energy of frames. Then, WDMM is formed by accumulating the input sequence in such a way that recent frames have more contribution than passed frames. It makes descriptor robust against temporal direction of gestures. As handcrafted features, HOG and LBP extracted from WDMM, are encoded by VLAD. These are then classified by the SLFN with ELM.

III. Proposed Method

The general overview of the proposed method is depicted in Figure 1. This paper aims at designing a robust representation for HGR. Then, by transforming the depth sequence into this representation, its target is labeling the video.

A. Problem Definition

Input videos of the proposed method are sequences of depth maps of hand, shown by $\{d_t|1 \leq t \leq T\}$. The output of the method is the label of each input video. In each sample, the subject is performing one meaningful hand gesture. First, the hand region is normalized to a fixed size (to cope with different hand sizes). To make the proposed method robust to the length of input videos (i.e., intra-class variation), the MTS method is introduced. Based on the key frame extraction process, the MST produces a long, middle, and short level of videos with different fixed numbers of frames from the original one.

Each depth frame is then projected onto three Cartesian planes to form 2D projected images. To take into account the temporal information, each sequence is divided into shorter clips. A temporal weighted version of DMM (WDMM) is proposed to compute one image from a sequence of depth frames. For describing each WDMM, it is divided into some patches. The HOG and LBP descriptors are employed to extract features from each patch of $WDMM^{l,v}_{c,p}$, which is the p^{th} patch of the WDMM computed from the c^{th} clip with the temporal level of l in view v. All of the features extracted from one sample are encoded by the VLAD encoding. The SLNF with ELM method is exploited for the classification.

B. Notations and Terms

Prior to presenting the proposed method, the main terms used in the rest of this paper are first presented in this subsection. The input video is defined as $\{d_t|1 \leq t \leq T\}$, where d_t is the t^{th} frame of the input sequence. In this paper, the input depth map is first converted to a binary image. In other words, each pixel of the depth map has a binary value

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology

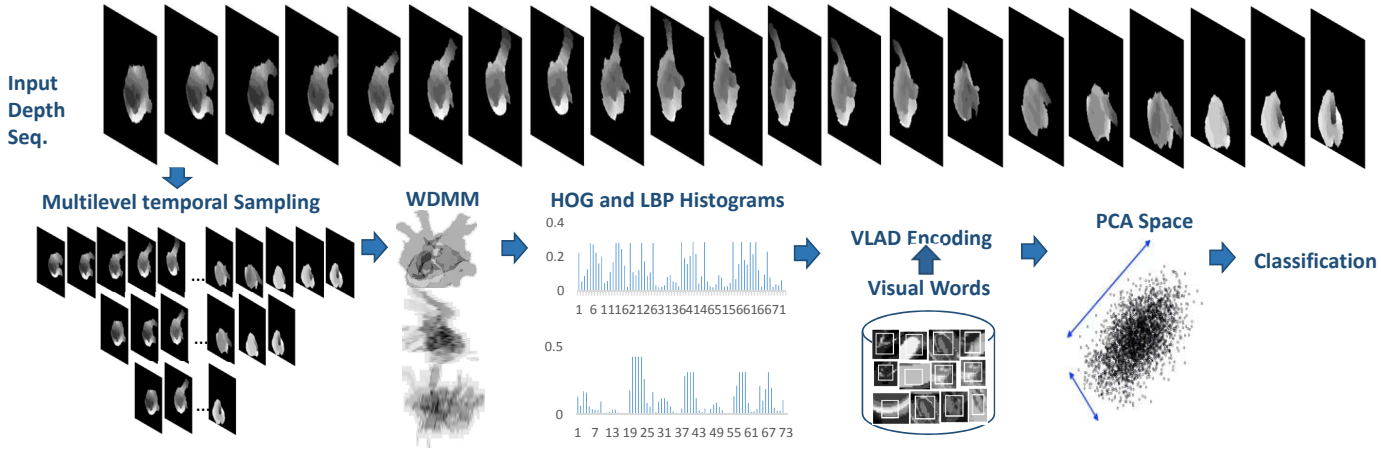


Fig. 1: General overview of proposed method.

of {0,1}. Therefore, d'_t can be defined as the complementary of d_t (in which, zeros become ones and ones become zeros). The motion energy of frame t is represented by E_t. Term $l \in \{long, middle, short\}$ is exploited to show the levels of temporal sampling. The projection views are indicated by $v \in \{top, side, bottom\}$. Also, N_c is the number of clips for each input video. The length of each clip is denoted by K. The feature vector extracted by the HOG and LBP are shown by $\mathbf{H}^v$ and $\mathbf{L}^v$, respectively. The final feature descriptors for each view of v after employing the VLAD and PCA are defined by $\mathbf{V}^v_{\mathbf{H}}$ and $\mathbf{V}^v_{\mathbf{L}}$ for HOG and LBP, respectively.

C. Multilevel Temporal Sampling

An important challenge in gesture recognition is the intra-class variation on account of different execution rates of the same gestures. To address this important problem, a new method based on MTS is proposed. A naive method could be down-sampling by selecting random frames. However, some relevant information in unselected frames might be lost. In order to mitigate this issue, the MTS method based on the motion energy of each frame is proposed. To do so, the motion energy of each frame is defined by accumulating the differences of each frame with its next frame over all pixel values, as $E_t = \sum_{i=1}^{N} (d_t(i) - d_{t+1}(i))^2$, where d_t is the t^{th} depth frame of the input video, N is the number of pixels in one frame, and E_t is the energy of that frame. By expanding this equation, the motion energy is $E_t = \sum_{i=1}^{N} (d_t^2(i) - d_t(i)d_{t+1}(i)) + \sum_{i=1}^{N} (d_{t+1}^2(i) - d_t(i)d_{t+1}(i))$. It is assumed that the value of each depth pixel is binary. Thus, $d_t^2(i) = d_t(i)$. Now, the motion energy will be

$$\begin{aligned} E_t &= \sum_{i=1}^{N} (d_t(i)(1 - d_{t+1}(i))) + \sum_{i=1}^{N} (d_{t+1}(i)(1 - d_t(i))) \\ &= \sum_{i=1}^{N} (d_t(i)\ d'_{t+1}(i)) + \sum_{i=1}^{N} (d_{t+1}(i)\ d'_t(i)), \end{aligned} \tag{1}$$

where d'_t is the complementary of d_t, in which, zeros become ones and ones become zeros.

There are two components in Equation 1 (i.e., $d_t(i)\ d'_{t+1}(i)$ and $d_{t+1}(i)\ d'_t(i)$), which can be considered as the backward and forward motion. What is important for gesture recognition is the movement of human body during the temporal dimension. If the background of frames is removed, the movement of human body can be introduced as the new parts of the space that are occupied by that human body in the next frame. In other words, the pixel values change from 0 to 1. Therefore, the forward motion is considered as the final motion energy function for each frame, as $E_t = \sum_{i=1}^{N} \left(d_{t+1}(i)\ d'_t(i)\right)$.

To sample video frames in different levels, it is crucial to select frames with relevant visual information (to discriminate different gestures and maximize the information contained in the original video). Here, input frames are sampled based on the change rate of motion energy $\Delta E = |E_t - E_{t+1}|$; i.e., the difference of the energy function of each frame and its next frame. Particularly, frames with higher body movements than their neighbors are selected.

In order to have a video with the fix length of M, the first and the last frames are first selected and then $M - 2$ frames with the highest ΔE values are sampled from the rest of the video. Here, three levels of *long*, *middle*, and *short* temporal samples are extracted from the original video, denoted by $l \in \{long, middle, short\}$, where the long level is the original video, the middle one contains 50% of the length of the original video, and the 30% of the input length is used to form the short level. It is worth mentioning that different number of levels were tested and three levels were empirically selected according their better results. The next steps of the proposed method are applied on each level, separately.

D. Weighted Depth Motion Map

Depth frames can be used as a part of the point cloud of the environment. For each depth video sequence, depth frames are first projected onto three orthogonal Cartesian planes. This projection forms the 2D projected images corresponding to the three projection views of *front*, *side*, and *top* denoted by $d^{l,v}$, where $v \in \{front, side, top\}$.

The DMM introduced in [7] captures information changes along the temporal dimension for each view. For each pixel, differences of depth values between two consecutive frames are calculated. The original DMM [7] is obtained by stacking the differences greater than a given threshold. The main

problem related to the original DMM is that the order of temporal information is lost. To address this problem, in the proposed method, two solutions are given. First, the WDMM is introduced. Based on this idea, the absolute differences of frame pixels are stacked by using a weighting method. In other words, pixel values near to the ending frames have more effect on the WDMM. Next, instead of computing the WDMM for the whole video sequence, the input is divided into N_c shorter clips with a fixed length of K. Then, WDMM is computed for all clips, separately. These clips are selected with 50% of overlap. Finally, WDMM is calculated as $WDMM = \sum_{t=1}^{K} |d_t^{l,v} - d_{t+1}^{l,v}|2^t$, where K is the length of clips. It is selected as 16, 8, and 4 for long, middle, and short videos, respectively. The final algorithm for extracting WDMM is shown in Algorithm 1. In Figure 2(a), some samples of WDMM for different levels of a front view are shown.

Algorithm 1 Extraction of WDMM

```
Input video = d_1, d_2, ..., d_T (T is the video length)
Output = WDMM_c^{l,v}
for l ∈ {long, middle, short} do
    for v ∈ {front, side, top} do
        for c=1 to N_c do
            WDMM_c^{l,v} = Σ_{t=(c-1)K/2+1}^{c K/2+K} |d_t^{l,v} - d_{t+1}^{l,v}| 2^t
        end for
    end for
end for
```

E. Descriptor

The HOG and LBP features are first extracted from the WDMM of each clip. Then, extracted features are mapped to a new feature space by the VLAD encoding approach. The PCA is then applied on encoded features to produce the dimension reduction final feature description of depth videos.

1) Feature Extraction: Patch-based HOG and LBP are two simple, yet effective, feature extraction methods. The HOG feature describes the local hand motion and appearance of WDMM. It computes histograms of spatial gradients by counting the occurrences of gradient orientations in localized portions of WDMMs. To do that, each WDMM is divided into small interconnected areas, of size 8×8, called a cell. The histogram of gradient directions, of size 9, is computed for all pixels within each cell by quantizing the range $[0, \pi]$ of gradient orientations. In a higher level, a WDMM is divided into P patches which are considered as four neighboring cells, denoted by $WDMM_{c,p}^{l,v}$ $(1 \leq p \leq P)$. Patches are selected with 50% of overlap. The patch feature is formed by concatenation of features from its contained cells. Therefore, it has a feature vector of length 36. For each input video, three matrices of HOG features are extracted for the three different views. Each HOG has a matrix of $\mathbf{H}^v \in \mathbb{R}^{3PN_c \times 36}$, where $v \in \{front, side, top\}$, P is the number of patches, N_c is the number of clips, 3 is the number of levels, and 36 is the length of each HOG feature vector. The rows of matrix $\mathbf{H}^v$ are related to the features extracted from $WDMM_{c,p}^{l,v}$. In Figure 2(b), the HOG description of three levels of temporal sampling is shown.

The LBP is an effective texture descriptor that has been used in various image processing and computer vision applications, thanks to its high computation speed and discriminative power. It can be considered as a temporal template for representing gestures and actions. The LBP describes the local texture pattern of an image by labeling all image pixels with a binary code. This binary code is calculated by comparing the gray value of each pixel (as a center) with its neighbors. For each pixel q, a set of m neighbors $\mathcal{N}(q, r)$ is defined such that these pixels are equally spaced on a circle of radius $r, (r > 0)$, with the center at q. In [34], the LBP of pixel q is calculated as $LBP(q) = \sum_{i=1}^{m} \Big((d(q_i') - d(q)) > 0\Big) 2^{i-1}$, where $q_i' \in \mathcal{N}(q, r)$ is the m^{th} neighbor around pixel q with a circle of radios r centered at q. In this work, $r = 1, 2$, and 3 are used. The LBP is depicted in Figure 2(c) for three levels of temporal sampling.

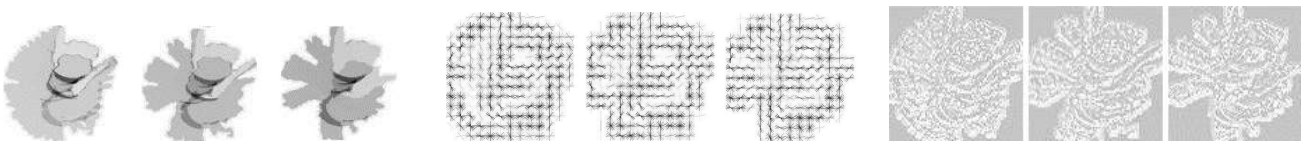
Like HOG, for LBP the input frame is divided into P patches. Patch features are formed by calculating the histogram of LBP codes of all pixels within it. For each input video, three matrices of LBP features are extracted for the three different views. Each LBP representation is a matrix of $\mathbf{L}^v \in \mathbb{R}^{3PN_c \times 59}$, where $v \in \{front, side, top\}$, P is the number of patches, N_c is the number of clips, 3 is the number of levels, and 59 is the length of each LBP feature vector. Rows of matrix $\mathbf{L}^v$ are related to the features extracted from $WDMM_{c,p}^{l,v}$ $(1 \leq p \leq P)$.

2) VLAD Encoding: For each WDMM of the clip, there are two kinds of separate feature matrices ($\mathbf{H}^v$ and $\mathbf{L}^v$). Each row of these matrices is an n-dimensional vector. To compute the final feature descriptor of the video, the VLAD encoding process is used (it is an efficient super vector encoding method). It can be considered as a kind of feature mapping exploited to transform the local features into a fixed size vector representations. The encoding is applied on HOG and LBP, separately. Details of VLAD encoding can be found in [10]. Some of visual words extracted for VLAD encoding are shown in Figure 3.

The normalized VLAD encoding of the three projection views are concatenated to form the video feature descriptor. The final feature descriptor is computed by performing PCA on the video feature. The algorithm of feature description is shown in Algorithm 2.

3) Classification: The SLFN with ELM method is used for classification [11]. The ELM randomly chooses hidden nodes and analytically determines the output weights of SLFNs. Gradient decent-based methods, which have been mainly used in the learning algorithm of feed-forward neural network, have two drawbacks. They are generally very slow and may easily converge to local minima. To overcome these problems, ELM [11] is used. It tends to reach the smallest norm of the weights while trying to minimize the training error. The learning speed of ELM is thousands of times faster than the traditional feed-forward network learning [11].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology



(a) (b) (c)

Fig. 2: a) Three levels of temporal sampling long, middle, and short (from left to right), b) HOG description for three levels of temporal sampling c) LBP description for three levels of temporal sampling.

Fig. 3: Some extracted visual words from hand regions.

Algorithm 2 Feature Description

Input = All sequences of $WDDM_c^{l,v}$ for one sample
Output = $Descriptor_{\mathbf{H}}$, $Descriptor_{\mathbf{L}}$
for $v \in \{front,\ side,\ top\}$ **do**
  for $l \in \{long,\ middle,\ short\}$ **do**
    for c = 1 **to** N_c **do**
      for p = 1 **to** P **do**
        $\mathbf{H}^v = HOG(WDMM_{c,p}^{l,v})$
        $\mathbf{L}^v = LBP(WDMM_{c,p}^{l,v})$
      end for
    end for
  end for
  $\mathbf{V}_{\mathbf{H}}^v = PCA(VLAD(H^v))$
  $\mathbf{V}_{\mathbf{L}}^v = PCA(VLAD(L^v))$
end for
$Descriptor_{\mathbf{H}} = \left[\mathbf{V}_{\mathbf{H}}^{top}, \mathbf{V}_{\mathbf{H}}^{front}, \mathbf{V}_{\mathbf{H}}^{side}\right]$
$Descriptor_{\mathbf{L}} = \left[\mathbf{V}_{\mathbf{L}}^{top}, \mathbf{V}_{\mathbf{L}}^{front}, \mathbf{V}_{\mathbf{L}}^{side}\right]$

IV. Experimental Results

The proposed HGR method is evaluated on four RGB-D datasets of MSR Gesture 3D, SKIG, and MSR Action 3D, and NTU. These are described in Section IV-A. The method is evaluated for different parameter values in Section IV-B. Performance of the method is compared against the state-of-the-art methods for RGB-D action recognition in Section IV-C.

A. Datasets

Four challenging datasets of MSR Gesture 3D, SKIG, MSR Action 3D, and NTU are utilized in the evaluation process.

1) MSR Gesture 3D Dataset: The MSR Gesture 3D dataset (Figure 4) [19] contains depth sequences of 12 dynamic *American sign language* (ASL) gestures of *bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j, and z*. Each gesture contains the segmented hand portion (above the wrist). It is performed by 10 subjects for twice or 3 times. There is no available RGB data for this dataset.

2) SKIG Dataset: The SKIG [6] is a hand gesture dataset which includes totally 2160 hand-gesture video sequences from six people, 1080 RGB sequences, and 1080 depth sequences. In this dataset (Figure 4), there are 10 categories of gestures *triangle (anti-clockwise), circle (clockwise), right and left, up and down, wave, hand signal Z, come-here, cross, pat, and turn around*. All these sequences are extracted through a Kinect sensor and the other two synchronized cameras.

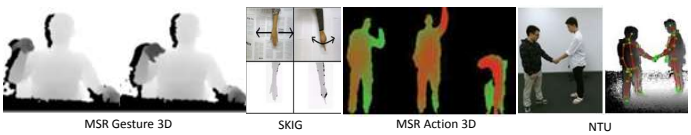


Fig. 4: Samples from the utilized datasets [19, 6, 35].

In order to increase the variety of recorded sequences, subjects are asked to perform three kinds of hand postures: fist, flat, and index. Furthermore, three different backgrounds (i.e., wooden board, paper with text, and white plain paper) and two illumination conditions (light and dark) are used in SKIG. Therefore, in total, there are 360 different gesture sequences accompanied by hand movement annotations for each subject.

3) MSR Action 3D Dataset: The MSR Action 3D Dataset [35] contains gaming actions (Figure 4). It consists of depth sequences of 20 actions of: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, and pick up and throw*; each performed by 10 subjects for twice or 3 times. The frame rate is 15 fps with the resolution of 320×240. The background of this dataset is removed. The most important challenge of this dataset is the inter-action similarities. It only contains depth videos.

4) NTU Dataset: The NTU dataset [37] (Figure 4), captured with Kinect (v2), is currently the largest available RGB-D action dataset, with more than 56000 sequences and 4 million frames. This dataset contains 60 action classes including daily actions, medical conditions, and pair actions. The sequences of skeletal information (25 body joints), RGB and depth frames are available for all samples. These actions are performed by 40 different people aged between 10 and 35, including both one-person daily actions (e.g., clapping, reading, writing) and two-person interactions (e.g., handshaking, hug, pointing). All samples are captured by three cameras which have been placed at different locations and view points. In total, there are 80 distinct viewpoints. The large intra-class and viewpoint variations make the dataset very challenging.

B. Parameter Setting

There are some parameters to be set in the proposed method. To set these parameters it is needed to use the validation set. As there is no prepared validation set for all datasets, 20% of training data is randomly selected as the validation set. All parameters are set by evaluation. From all utilized

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology

parameters, the number of visual words and the number of PCA components significantly affect the final accuracy. These parameters are evaluated in this Section.

1) Visual Words: One of the used parameters is the number of visual words. This parameter is evaluated with three kinds of features: HOG, LBP, and the combination of both. Different values from 8 to 128 have been tested for this experiment. Figure 5(a-c) shows the effect of this parameter on the validation set of three datasets of MSR Gesture 3D, SKIG, MSR Action 3D, respectively. The best accuracy was achieved by 25, 30, and 70 (Figure 6(a)) number of visual words for MSR Gesture 3D, SKIG, and MSR Action 3D, respectively. The accuracy of the proposed method for different values of the number of visual words is depicted in Figure 5(d) for NTU dataset. It can be seen that the best number of visual words is 70 for this dataset.

The number of visual words (i.e., dictionary size) is a factor that affects the performance of action recognition. Each visual word in a dictionary describes a kind of local feature included in input data. Therefore, a small number of visual words may lack the discriminative power (since two kinds of local feature may be assigned into the same visual word even if they are not similar to each other). On the other hand, a large number of visual words can completely deteriorate the performance, because it is less generalizable, more sensitive to noise, and contains extra processing overhead. Since visual words describe the conceptual content of images, the number of visual words is highly dependent on the used dataset. For representing data with more complex information, larger number of contextual visual words is needed.

Among four utilized datasets in this work, the extracted WDMMs from MSR Action 3D and NTU have more local features since these datasets contain the whole human body in all frames. Therefore, a larger number of visual words is expected for these datasets than the ones based on hand gestures. Figure 5(d) and 5(a) show that the best accuracy for these datasets is achieved by 70 as the dictionary size. The MSR Gesture 3D and SKIG datasets have 12 and 10 categories of gestures, respectively. Although the number of classes for the MSR Gesture 3D is larger than the SKIG, gestures from different classes are more similar in the MSR Gesture 3D rather than in the SKIG. Moreover, frames of the MSR Gesture 3D include only the hand part of the body while the hand and forearm are visible in samples of the SKIG dataset. Hence, for the SKIG, more visual words are needed to discriminate gestures due to more image content and inter-class similarities.

These evaluations also show that the proposed method achieves the best result by using the combination of the HOG and LBP features. The rest of the experiments are performed by utilizing $25, 30, 70$, and 70 as the number of visual words for the MSR Gesture 3D, SKIG, MSR Action 3D, and NTU datasets, respectively.

In all experiments, the accuracy of the method is depicted for the range of $[25, 128]$ of the number of visual words. By increasing the number of visual words to larger numbers (larger than 128), the accuracy does not change for a while. This increment makes the method more sparse and causes additional processing overhead. After a while, the accuracy decreases and overfitting occurs.

2) PCA: The number of components selected by PCA is another parameter that is needed to be set. Figure 6(b) shows the overall accuracy of the proposed method for different number of PCA components on the validation set. By increasing the number of PCA axes from 70 to 130, the accuracy increases for all datasets. The method is also evaluated with different number of PCA components, from 100 to 4000, for the NTU. Since this dataset has more number of classes, the length of feature vectors may require to be larger. By increasing this number from 100 to 1000, the accuracy of the method grows. Then, for the range of 1000 to 3000, the accuracy is almost the same. After 130, for the first three datasets, and 3000 for the NTU, the accuracy decreases or does not change. This might be related to the fact that the highly dimensional space of the feature vector increases the possibility of overfitting. From Figure 6(b), it can be seen that 130 is the best value for the first three datasets, and 2000 is the best size for the NTU. The rest of results are achieved by this size of features.

C. Performance Comparison

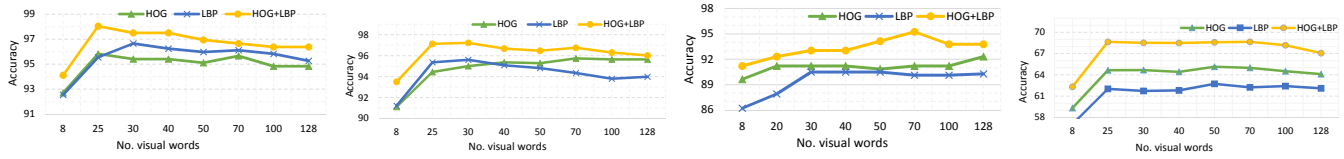
In order to compare the accuracy of the proposed method with deep learning-based method, the CNN is utilized to classify the WDMM. To do that, three networks (AlexNet) are trained for three views. Since there is no pre-trained network with the WDMM as the input data, CNNs are finetuned from a pre-trained network on the ImageNet. Weighted score averaging is used to combine the results from three views. It is worth mentioning that some features are extracted from different layers of the CNN and are replaced with handcrafted features. The result has no significant difference with an end-to-end network.

In Figure 7(a), the confusion matrix of the proposed method for MSR Gesture 3D is depicted. As this figure shows, the proposed method is able to correctly classify most of the gestures within 9 categories for this dataset.

Table I lists the accuracy obtained by different methods on the MSR Gesture 3D dataset. The achieved accuracy of the combination of HOG and LBP features is better than processing each feature separately. It can also be seen that the accuracy of the LBP is higher than that of the HOG, with a small margin. In fact, as HOG is a gradient-based feature extraction method, it considers the changing rate of values of pixels with their neighbors for different orientations, separately. However, LBP features are based on comparison of pixels with all of their neighbors in all directions at the same time. Therefore, LBP is more powerful than HOG in this case. Among all methods listed in Table I, the proposed method achieves the best result. Note that the combination of HOG and LBP features outperforms the CNN.

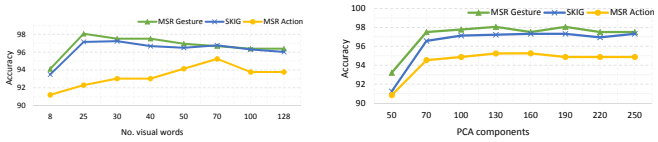
The confusion matrix on the SKIG dataset is shown in Figure 7(b). Comparison of the confusion matrix on MSR Gesture 3D and SKIG dataset shows that the SKIG dataset has more inter-class similarities. The per class accuracy of only 2 classes of gestures is 100% for SKIG. Gesture categories in SKIG datasets are more similar to each other than MSR Gesture 3D dataset. This inter-class similarity, in most classes, causes one or more misclassified samples.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology



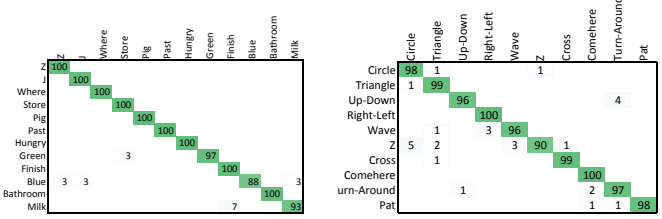
(a) (b) (c) (d)

Fig. 5: Accuracy vs the number of visual words for: a) MSR Gesture 3D, b) SKIG, c) NTU, and d)MSR Action 3D Datasets.



(a) (b)

Fig. 6: Accuracy vs the number of a) visual words, b) PCA components, for three datasets.



(a) (b)

Fig. 7: Confusion matrix on a) MSR Gesture 3D, b) SKIG.

In Table II, the performance of the proposed method is compared with that of the state-of-the-art methods on the SKIG dataset. For the SKIG dataset, the LBP results in higher accuracy than HOG. The accuracy of the combination of LBP and HOG features is higher than most of other state-of-the-art methods except one. The method proposed by Molchanov et al. [30] is a deep learning-based method. It utilizes both RGB and depth data while the proposed method only uses the depth data. For this dataset, also, the combination of HOG and LBP features outperforms the CNN.

To show the generalization capabilities of the proposed method, an action recognition dataset is also evaluated. In Figure 8, the confusion matrix on MSR Action 3D dataset is shown. The per class accuracy for 13 actions is 100%.

Table III compares the performance of the proposed method with that of the existing state-of-the-art methods. Unlike other previous datasets, the accuracy of HOG features is better than LBP for this dataset. The reason might be related to the

TABLE I: Performance comparison on MSR Gesture 3D.

Method	Year	Modality	Accuracy
ROP[36]	2012	Depth	88.50
DMM + HOG[7]	2012	Depth	89.20
HON4D[22]	2013	Depth	92.45
H3D Facets[39]	2015	Depth	95.0
DMM+ LBP[8]	2015	Depth	95.0
Subspace encoding[40]	2016	Depth	95.50
HKD [20]	2016	Depth	96.09
Histogram data[41]	2017	Depth	94.70
SSTKDes[21]	2017	Depth	97.02
Active Incremental Learning[42]	2017	Depth	91.25
Extended SNV [23]	2017	Depth	94.74
3DHoTs[41]	2017	Depth	94.7
Proposed CNN	2018	Depth	97.21
Proposed HOG	2018	Depth	96.22
Proposed LBP	2018	Depth	96.52
Proposed HOG + LBP	**2018**	**Depth**	**98.05**

TABLE II: Performance comparison on SKIG.

Method	Year	Modality	Accuracy
Discriminative Rep.[6]	2013	RGB-D	88.7
Shape Model[43]	2014	RGB-D	96.0
4DCov [44]	2014	Depth	93.8
Binary Rep.[45]	2016	RGB-D	93.7
Depth context [46]	2016		95.37
3D CNN[30]	2016	RGB-D	98.6
LSTM [47]	2017	Depth	91.30
Fusion of deep[47]	Depth	2017	93.3
Proposed CNN	2018	Depth	96.11
Proposed HOG	2018	Depth	95.0
Proposed LBP	2018	Depth	95.60
Proposed HOG + LBP	**2018**	Depth	**97.31**

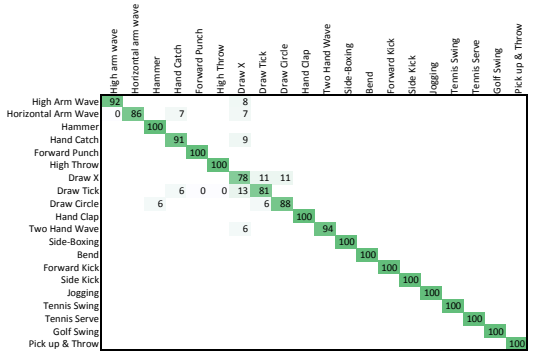


Fig. 8: Confusion matrix on MSR Action 3D dataset.

fact that MSR Action 3D dataset contains full body actions while other datasets include only hand gestures. For gesture datasets, the relative position of different parts of hand is very important. Therefore, the LBP can describe these positions better than HOG. In the action dataset, the motion and shape of the boundary of human body are more important in the temporal dimension. Therefore, HOG which contains gradient-based features, can handle this situation better than LBP.

Table III shows that the accuracy of the proposed method which is higher than most of the state-of-the-art methods. It is even comparable to the best accuracy which is achieved by [21]. For this dataset, there is a high gap between the accuracy of CNN and the combination of HOG and LBP. This fact shows that for a small dataset, handcrafted methods still outperform deep learning-based methods.

In order to evaluate the proposed method on large benchmark datasets, the NTU (the largest RGB-D action recognition dataset to date) is exploited. There are two types of evaluation procedures for this dataset, namely the cross-subject and cross-view. In the cross-subject evaluation, 20 subjects are used as the train set and the remaining subjects are reserved for test. On the other hand, in the cross-view evaluation, two views are utilized as the train and one view is used for test. Other depth-based approaches that give the evaluation results on the NTU, only publish the cross-subject accuracy. In other words, only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology

TABLE III: Performance comparison on MSR Action 3D.

Method	Year	Modality	Accuracy
Bag of 3D Points[35]	2010	Depth	74.70
ROP[36]	2012	Depth	86.20
HON4D[22]	2013	Depth	88.89
DMM+ LBP[8]	2015	Depth	94.9
Subspace encoding[40]	2016	Depth	94.06
Active joints[48]	2017	Skeleton	84.72
SSTKDes[21]	2017	Depth	95.60
3DHoTs[41]	2017	Depth	95.2
Extended SNV [23]	2017	Depth	93.45
Trust gate [49]	2017	Skeleton	94.8
ST-NBNN[50]	2017	Skeleton	94.8
Proposed CNN	2018	Depth	90.00
Proposed HOG	2018	Depth	91.94
Proposed LBP	2018	Depth	91.57
Proposed HOG + LBP	**2018**	**Depth**	**95.24**

TABLE IV: Performance comparison on NTU.

Method	Year	Modality	Accuracy
SNV [23]	2014	Depth	31.82
HON4D [22]	2013	Depth	30.56
FTP Dynamic Skeletons [55]	2015	Skeleton	60.23
Trust gate [49]	2017	Skeleton	69.2
ConvLSTM [33]	2017	RGB-D	66.2
MTLN [56]	2017	Skeleton	79.57
Lie groups [54]	2017	Skeleton	61.37
Two-stream RNN [53]	2017	Skeleton	71.3
GCA-LSTM [59]	2017	Skeleton	74.4
SkeletonNet [60]	2017	Skeleton	75.94
SLP-TEP [51]	2017	Depth	58.22
DSSCA-SSLM [52]	2017	Depth	74.86
Proposed CNN	**2018**	Depth	**75.16**
Proposed HOG	2018	Depth	65.14
Proposed LBP	2018	Depth	62.73
Proposed HOG+LBP	2018	Depth	68.66

skeleton-based approaches provide the cross-view evaluation results. Following other depth-based papers, the cross-subject evaluation results are reported.

Table IV compares the performance of the proposed method on the NTU dataset with that of the existing state-of-the-art methods. This table shows that the HOG descriptor works better than the LBP. Like the MSR Action 3D, the NTU is an action dataset. Therefore, the motion and shape of the boundary of human body are more important in the temporal dimension. As a result, the HOG descriptor works better than the LBP. Moreover, it can be seen that for this dataset, the CNN results in higher accuracy than the combination of HOG and LBP, while for the other datasets handcrafted features work better than the CNN. This evaluation demonstrates that with the same input data, only with large enough data, deep models are superior to handcrafted features for classification purposes. In other words, with a large amount of input data, deep models are able to learn better weights for classification (but when the input data is not sufficient, compared to handcrafted features, deep models are not successful enough for classification purposes).

In comparison with RGB-D-based approaches [23, 22, 33, 51], the proposed method with both handcrafted features and CNN achieves higher accuracies, and the results of deep-based methods are comparable with [52]. Among the skeleton-based methods, the achieved result of the proposed method is higher than [49, 53, 54, 55]. There are some recent skeleton-based approaches [56, 57, 58] whose results are about $79-80\%$. In those approaches some complex deep networks or the combination of different kinds of those models [33, 56] are proposed to classify actions. Therefore, a large number of weights needs to be learned. On the other hand, compared to other deep learning-based methods, the best accuracy of the proposed method is achieved by a simple CNN model (i.e., AlexNet). Therefore, other deep learning-based approaches have more computational complexity than the proposed method. This fact, in turn, demonstrates that the input data to the CNN model is discriminate enough for gesture and action classification.

Moreover, the skeletal data is extracted from depth map, and it has more semantic information than the value of pixels in depth maps. However, it has also some limitations. Many of usual gestures and actions are involved with the interaction of the body with other objects. In some cases, appearances of body parts and objects in the environment provide discriminative information. As such, the skeletal data is insufficient to distinguish gestures and actions which involve human-object interactions. For instance, the skeleton-based methods might fail when the gestures or actions with very similar skeleton movements interact with totally different objects.

Moreover, in skeleton-based models, the estimation of body or hand joints is unreliable (or even fails) in some real-life cases including: i) when the resolution of the depth map is not high enough, ii) in presence of occlusion or self-occlusion, iii) when the subject touches the background, and iv) in outdoor environments. With inaccurate body joints, the intra-class variations increase in gestures or actions. Furthermore, the computational cost of extraction of skeleton in multi-person frames can be very high.

D. Multilevel Temporal Sampling Analysis

One of the most important challenges of gesture recognition is different execution rates. In other words, the video length of the same hand gestures is variable for different subjects. The histograms of length for the three datasets of MSR Gesture 3D, SKIG, and MSR Action 3D are depicted in Figure 9.

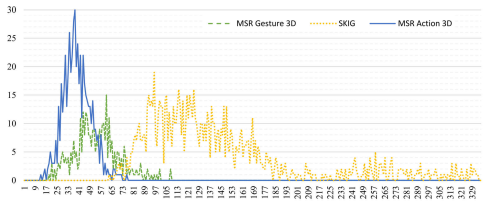


Fig. 9: Histogram of video length for all dataset.

It can be seen that there is a large variation in the length of videos. The idea of MTS yields the method to achieve descriptors which are invariant to video length. Table V lists the accuracy achieved by using different temporal levels on MSR Action 3D dataset. The accuracy of the case which uses all three temporal levels is the best. It also shows that using one or two temporal levels of sampling is not enough.

E. Computational Complexity Analysis

The proposed method is implemented in Matlab 2016a on a Core i7 740Q system. Table VI shows the percentage of time spent on each step of the proposed method. Extraction of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2855416, IEEE Transactions on Circuits and Systems for Video Technology

TABLE V: Performance comparison on MSR Action 3D dataset with different levels of temporal sampling.

Multilevel temporal sampling	Accuracy
Long	93
Long + Short	94.20
Long+Long+Long	93.6
Long+ Middle+ Short	95.24

TABLE VI: Average run-time on all datasets.

Step	Run Time (%)
MTS	0.8
HOG	0.9
LBP	0.93
VLAD Encoding	97.37

MTS contains two steps of computing the motion energy for each frame of a sequence and then sorting them to select the frames with the highest motion energy. The first part is an order of $O(N)$ (N is the number of pixels within frame) and the second part is an order of $O(Tlog(T))$ (T is the video length). Given that in our case $T \ll N$, the whole process of MTS is an order of $O(N)$. Extraction of both LBP and HOG is an order of $O(N)$. In Table VI, it can be seen that the three steps of MTS, HOG, and LBP take almost the same time to compute. The VLAD encoding includes two parts of creating dictionary of visual words and then assigning each sample to the visual words. Computing visual words (i.e., k-means) takes the most part of time with 97.37%. The running time of k-means (Lloyd's algorithm) is $O(nk)$ where n is the whole number of samples of the dataset, and k is the number of visual words (8 to 128). Assigning samples to the visual words is an order of the whole number of samples in dataset $O(n)$.

F. Effect of Combining Different Features

Utilizing the combination of HOG and LBP features improves the accuracy for all three datasets. The *histogram of the second-order gradient* (HOG2) [38], *gray level co-occurrence matrix* (GLCM) [61], and CNN are also evaluated on MSR action 3D dataset to compare the accuracy achieved by these descriptors with other existence descriptors. Figure 10 shows the accuracy of different combinations of these descriptors. It can be seen that the combination of HOG and LBP features outperforms other methods.

V. Conclusion

A novel and effective method for human gesture recognition in sequences of depth maps was proposed. First, three levels of temporal samples (long, middle, and short) were computed from the input video to extract videos with high motion information. Next, each depth video was projected onto three orthogonal Cartesian views. Those videos were divided into some shorter clips with overlap and fixed lengths. For each clip, the proposed WDMM was computed as the accumulated weighted absolute difference of consecutive frames. The HOG and LBP descriptors were extracted from all WDMMs. At the end, the final descriptor was formed by applying the PCA on the VLAD encoding of extracted features. Then, the SLFN method with ELM was used as the classifier. It was shown that MTS yields the method to be robust to different execution rates of performing gestures, resulting in increasing the accuracy of the gesture recognition rate. Moreover, the LBP and HOG and also their combination were evaluated on four benchmark datasets. For gesture datasets, the LBP achieved better results and for the action dataset the HOG outperformed other methods. Finally, the experimental results showed the efficiency and superiority of the proposed method when compared to the state-of-the-art methods on the MSR Gesture 3D, SKIG, and MSR Action 3D datasets. We also showed competitive results on NTU action dataset. As our future work, the fusion of skeleton and depth features will be considered to analyze their complementary impact on improving the overall recognition performance.

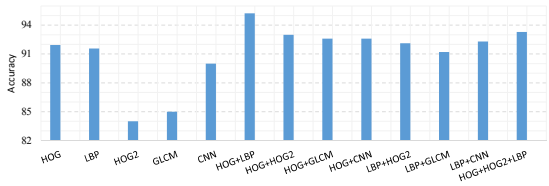


Fig. 10: Comparison of different combinations of descriptors.

Acknowledgment

This work was partially supported by the Spanish project TIN2016-74946-P (MINECO/FEDER, UE) and CERCA Programme / Generalitat de Catalunya. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

References

[1] Hang Zhou and Qiuqi Ruan. A real-time gesture recognition algorithm on video surveillance. In *Signal Processing, 2006 8th International Conference on*, volume 3. IEEE, 2006.

[2] Juan Wachs, Helman Stern, Yael Edan, Michael Gillam, Craig Feied, Mark Smith, and Jon Handler. A real-time hand gesture interface for medical visualization applications. *Applications of Soft Computing*, pages 153–162, 2006.

[3] Utpal V Solanki and Nilesh H Desai. Hand gesture based remote control for home appliances: Handmote. In *Information and Communication Technologies, 2011 World Congress on*, pages 419–423. IEEE, 2011.

[4] Siddharth S Rautaray and Anupam Agrawal. Real time gesture recognition system for interaction in dynamic environment. *Procedia Technology*, 4:595–599, 2012.

[5] Wei Lu, Zheng Tong, and Jinghui Chu. Dynamic hand gesture recognition with leap motion controller. *IEEE Signal Processing Letters*, 23(9):1188–1192, 2016.

[6] Li Liu and Ling Shao. Learning discriminative representations from rgb-d video data. In *IJCAI*, volume 4, page 8, 2013.

[7] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1057–1060. ACM, 2012.

[8] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Action recognition from depth sequences using depth motion maps-based local binary patterns. In *Applications of Computer Vision, 2015 IEEE Winter Conference on*, pages 1092–1099. IEEE, 2015.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition Conference on*, pages 3304–3311. IEEE, 2010.

[11] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[12] Hong Cheng, Lu Yang, and Zicheng Liu. Survey on 3d hand gesture recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1659–1673, 2016.

[13] Maryam Asadi-Aghbolaghi, Albert Clapés, Marco Bellantonio, Hugo Jair Escalante, Víctor Ponce-López, Xavier Baró, Isabelle Guyon, Shohreh Kasaei, and Sergio Escalera. Deep learning for action and gesture recognition in image sequences: A survey. In *Gesture Recognition*, pages 539–578. Springer, 2017.

[14] Maryam Asadi-Aghbolaghi, Albert Clapés, Marco Bellantonio, Hugo Jair Escalante, Víctor Ponce-López, Xavier Baró, Isabelle Guyon, Shohreh Kasaei, and Sergio Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *AFG 12th IEEE International Conference on*, pages 476–483. IEEE, 2017.

[15] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.

[16] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. 3d hand gesture recognition by analysing set-of-joints trajectories. In *International Conference on Pattern Recognition/UHA3DS 2016 workshop*, 2016.

[17] E Escobedo-Cardenas and G Camara-Chavez. A robust gesture recognition using hand local data and skeleton trajectory. In *Image Processing, 2015 IEEE International Conference on*, pages 1240–1244. IEEE, 2015.

[18] Mahmoud Elmezain, Ayoub Al-Hamadi, Saira Saleem Pathan, and Bernd Michaelis. Spatio-temporal feature extraction-based hand gesture recognition for isolated american sign language and arabic numbers. In *Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on*, pages 254–259. IEEE, 2009.

[19] Alexey Kurakin, Zhengyou Zhang, and Zicheng Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference, Proceedings of the 20th European*, pages 1975–1979. IEEE, 2012.

[20] Yu Kong, Behnam Satarboroujeni, and Yun Fu. Learning hierarchical 3d kernel descriptors for rgb-d action recognition. *Computer Vision and Image Understanding*, 144:14–23, 2016.

[21] Maryam Asadi-Aghbolaghi and Shohreh Kasaei. Supervised spatio-temporal kernel descriptor for human action recognition from rgb-depth videos. *Multimedia Tools and Applications*, pages 1–21, 2017.

[22] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[23] Xiaodong Yang and YingLi Tian. Super normal vector for human activity recognition with depth cameras. *IEEE transactions on pattern analysis and machine intelligence*, 39(5):1028–1039, 2017.

[24] Byeongkeun Kang, Subarna Tripathi, and Truong Q Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *Pattern Recognition, 3rd IAPR Asian Conference on*, pages 136–140. IEEE, 2015.

[25] Jonathan Wu, Prakash Ishwar, and Janusz Konrad. Two-stream cnns for gesture-based verification and identification: Learning user style. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 42–50, 2016.

[26] Pichao Wang, Wanqing Li, Zhimin Gao, Jing Zhang, Chang Tang, and Philip O Ogunbona. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, 46(4):498–509, 2016.

[27] Maryam Asadi-Aghbolaghi, Hugo Bertiche, Vicent Roig, Shohreh Kasaei, and Sergio Escalera. Action recognition from rgb-d data: Comparison and fusion of spatio-temporal handcrafted features and deep strategies. In *International Conference on Computer Vision Workshop*, pages 3179–3188, 2017.

[28] Pichao Wang, Wanqing Li, Zhimin Gao, Yuyao Zhang, Chang Tang, and Philip Ogunbona. Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. In *Proc. Comput. Vis. Pattern Recognit.*, pages 1–10, 2017.

[29] Yunan Li, Qiguang Miao, Kuan Tian, Yingying Fan, Xin Xu, Rui Li, and Jianfeng Song. Large-scale gesture recognition with a fusion of rgb-d data based on saliency theory and c3d model. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[30] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[31] Natalia Neverova, Christian Wolf, Giulio Paci, Giacomo Sommavilla, Graham Taylor, and Florian Nebout. A multi-scale approach to gesture detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 484–491, 2013.

[32] Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song, Syed Afaq Shah, and Mohammed Bennamoun. Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3120–3128, 2017.

[33] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. *arXiv preprint arXiv:1701.01821*, 2, 2017.

[34] Di Huang, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):765–781, 2011.

[35] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops, Computer Society Conference on*, pages 9–14. IEEE, 2010.

[36] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3d action recognition with random occupancy patterns. In *Computer vision–ECCV 2012*, pages 872–885. Springer, 2012.

[37] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. *arXiv preprint arXiv:1604.02808*, 2016.

[38] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and hog2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 465–470, 2013.

[39] Chenyang Zhang and Yingli Tian. Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition. *Computer Vision and Image Understanding*, 139:29–39, 2015.

[40] Chengwu Liang, Enqing Chen, Lin Qi, and Ling Guan. 3d action recognition using depth-based feature and locality-constrained affine subspace coding. In *Multimedia, 2016 IEEE International Symposium on*, pages 261–266. IEEE, 2016.

[41] Baochang Zhang, Yun Yang, Chen Chen, Linlin Yang, Jungong Han, and Ling Shao. Action recognition using 3d histograms of texture and a multi-class boosting classifier. *IEEE Transactions on Image Processing*, 26(10):4648–4660, 2017.

[42] Rocco De Rosa, Ilaria Gori, Fabio Cuzzolin, and Nicolò Cesa-

Bianchi. Active incremental recognition of human activities in a streaming context. *Pattern Recognition Letters*, 99:48–56, 2017.

[43] Pham Thanh Tung and Ly Quoc Ngoc. Elliptical density shape model for hand gesture recognition. In *Proceedings of the Fifth Symposium on Information and Communication Technology*, pages 186–191. ACM, 2014.

[44] Pol Cirujeda and Xavier Binefa. 4dcov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences. In *3D Vision, 2014 2nd International Conference on*, volume 1, pages 657–664. IEEE, 2014.

[45] Mengyang Yu, Li Liu, and Ling Shao. Structure-preserving binary representations for rgb-d action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1651–1664, 2016.

[46] Mengyuan Liu and Hong Liu. Depth context: A new descriptor for human activity recognition by using sole depth sequences. *Neurocomputing*, 175:747–758, 2016.

[47] Ling Shao, Ziyun Cai, Li Liu, and Ke Lu. Performance evaluation of deep feature learning for rgb-d image/video classification. *Information Sciences*, 385:266–283, 2017.

[48] Ahmad KN Tehrani, Maryam Asadi-Aghbolaghi, and Shohreh Kasaei. Skeleton-based human action recognition-a learning method based on active joints. In *VISIGRAPP*, pages 303–310, 2017.

[49] Jun Liu, Amir Shahroudy, Dong Xu, Alex Kot Chichung, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[50] Junwu Weng, Chaoqun Weng, and Junsong Yuan. Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4171–4180, 2017.

[51] Xiaopeng Ji, Jun Cheng, Dapeng Tao, Xinyu Wu, and Wei Feng. The spatial laplacian and temporal energy pyramid representation for human action recognition using depth sequences. *Knowledge-Based Systems*, 122:64–74, 2017.

[52] Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[53] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *e Conference on Computer Vision and Patern Recognition*, 2017.

[54] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 6099–6108. IEEE computer Society, 2017.

[55] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Computer Vision and Pattern Recognition, 2015 IEEE Conference on*, pages 5344–5352. IEEE, 2015.

[56] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4570–4579. IEEE, 2017.

[57] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[58] Hongsong Wang and Liang Wang. Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection. *IEEE Transactions on Image Processing*, 27(9):4382–4394, 2018.

[59] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, 2017.

[60] Qiuhong Ke, Senjian An, Mohammed Bennamoun, Ferdous Sohel, and Farid Boussaid. Skeletonnet: Mining deep part features for 3-d action recognition. *IEEE signal processing letters*, 24(6):731–735, 2017.

[61] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

Reza Azad was born in Ardabil, Iran, in 1989. He is currently a M.Sc. student at Sharif University of Technology, Tehran, Iran. His research interests include deep learning, computer vision and human computer interaction.

Maryam Asadi-Aghbolaghi received both B.Sc. and M.Sc. degrees in Computer Engineering from Iran University of Science and Technology, Tehran, Iran in 2008 and 2011, respectively. Currently, she is a PhD student at Sharif University of Technology, Tehran, Iran. Her research interests include human action recognition, ambient intelligence, 3D computer vision, machine learning, and data mining.

Shohreh Kasaei (M'05–SM'07) received the B.Sc. degree from the Department of Electrical and Computer Engineering (ECE), Isfahan University of Technology, Iran, in 1986. She then received the M.Sc. degree from Graduate School of Engineering and Science, Department of Electrical and Electronics Engineering, University of the Ryukyus, Japan, in 1994, and the Ph.D. degree from Signal Processing Research Center, School of Electrical Engineering and Computer Science (EECS), Queensland University of Technology (QUT), Australia, in 1998. She was awarded as the best graduate student in engineering faculties of University of the Ryukyus, in 1994, the best Ph.D. student studied in overseas by the ministry of Science, Research, and Technology of Iran, in 1998, and as a distinguished researcher of Sharif University of Technology (SUT), in 2002 and 2010, where she is currently a full professor. She is the director of Image Processing Lab (IPL). Her research interests are in image/video processing and 3D computer vision with primary emphasis on: self-driving cars, 3D reconstruction and graphical element addition in dynamic sports scenes, 3D dynamic pose estimation, 3D dynamic human action recognition, 3D model building, 3D object tracking, 3D semantic scene understanding, multi-resolution texture analysis, scalable video coding, image retrieval, video indexing, face recognition, hyperspectral change detection, video restoration, and fingerprint authentication.

Sergio Scalera obtained the P.h.D. degree on Multi-class visual categorization systems at Computer Vision Center, UAB. He obtained the 2008 best Thesis award on Computer Science at Universitat Autnoma de Barcelona. He leads the Human Pose Recovery and Behavior Analysis Group at UB, CVC, and the Barcelona Graduate School of Mathematics. He is an associate professor at the Department of Mathematics and Informatics, Universitat de Barcelona. He is an adjunct professor at Universitat Oberta de Catalunya, Aalborg University, and Dalhousie University. He has been visiting professor at TU Delft and Aalborg Universities. He is also a member of the Computer Vision Center at UAB. He is series editor of The Springer Series on Challenges in Machine Learning. He is vice-president of ChaLearn Challenges in Machine Learning, leading ChaLearn Looking at People events. His research interests include, between others, statistical pattern recognition, affective computing, and human pose recovery and behavior understanding, including multi-modal data analysis.