




Article

Exploring 3D Human Action Recognition Using STACOG on Multi-View Depth Motion Maps Sequences

Mohammad Farhad Bulbul ¹ , Sadiya Tabussum ¹, Hazrat Ali ² , Wenli Zheng ³, Mi Young Lee ^{4,*} and Amin Ullah ^{4,5,*} 

¹ Department of Mathematics, Jashore University of Science and Technology, Jashore 7408, Bangladesh; farhad@just.edu.bd (M.F.B.); sadiya.just.bd@gmail.com (S.T.)

² Department of Electrical and Computer Engineering, Abbottabad Campus, COMSATS University Islamabad, Abbottabad 22060, Pakistan; hazrat.ali@cuatd.edu.pk

³ School of Science, Xi'an Shiyou University, Xi'an 710065, China; wlzheng@xsyu.edu.cn

⁴ Intelligent Media Laboratory, Department of Software, Sejong University, Seoul 143-747, Korea

⁵ CORIS Institute, Oregon State University, Corvallis, OR 97331, USA

* Correspondence: miylee@sejong.ac.kr (M.Y.L.); qamin3797@sju.ac.kr (A.U.)

Abstract: This paper proposes an action recognition framework for depth map sequences using the 3D Space-Time Auto-Correlation of Gradients (STACOG) algorithm. First, each depth map sequence is split into two sets of sub-sequences of two different frame lengths individually. Second, a number of Depth Motion Maps (DMMs) sequences from every set are generated and are fed into STACOG to find an auto-correlation feature vector. For two distinct sets of sub-sequences, two auto-correlation feature vectors are obtained and applied gradually to L_2 -regularized Collaborative Representation Classifier (L_2 -CRC) for computing a pair of sets of residual values. Next, the Logarithmic Opinion Pool (LOGP) rule is used to combine the two different outcomes of L_2 -CRC and to allocate an action label of the depth map sequence. Finally, our proposed framework is evaluated on three benchmark datasets named MSR-action 3D dataset, DHA dataset, and UTD-MHAD dataset. We compare the experimental results of our proposed framework with state-of-the-art approaches to prove the effectiveness of the proposed framework. The computational efficiency of the framework is also analyzed for all the datasets to check whether it is suitable for real-time operation or not.

Keywords: 3D action recognition; depth motion maps; 3D auto-correlation features; decision fusion; Regularized Collaborative Representation Classifier (CRC)



Citation: Bulbul, M.F.; Tabussum, S.; Ali, H.; Zheng, W.; Lee, M.Y.; Ullah, A. Exploring 3D Human Action Recognition Using STACOG on Multi-View Depth Motion Maps Sequences. *Sensors* **2021**, *21*, 3642. <https://doi.org/10.3390/s21113642>

Academic Editor: Francisco Florez-Revuelta

Received: 10 March 2021

Accepted: 25 April 2021

Published: 24 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Human action recognition is one of the most challenging tasks in the area of artificial intelligence and has obtained attention due to widespread real-life applications, which extend from robotics to human-computer interface, automated surveillance system, healthcare monitoring, etc. [1–3]. Human actions are composed of contemporary behaviors of human body parts. The objective of human action recognition is to recognize actions automatically from an unlabeled video [4,5]. To capture human actions, there are two broad categories of devices based on wearable sensors and video sensors. In the prior, using these apparatuses many research works have been completed in the area of action recognition. To recognize wearable sensor-based actions, multiple sensors are connected to the human body. To obtain action information, most of the researchers have used different sensors such as accelerometers, gyroscopes, and magnetometers [6–8]. These wearable sensors are used in the healthcare system, worker monitoring, interactive gaming, sports, etc. However, they are not acceptable in all the domains of action recognition, for example in the automatic surveillance system. It is far from convenient for humans (especially patients) to wear the sensors for a long time and relatively it is difficult in cases of energy costs. Wearable sensors can have health risks. For those carrying smartphones, laptops, and tablets, wearable sensor increases

exposure to radio wave. Although the use of multiple sensors increases recognition accuracy, it has limitations for real-life applications because of increased associated complexity and the cost of the total procedure. Because of the difficulties of wearable sensors, video sensors such as RGB cameras are used to recognize the action. RGB images give restricted 2D data as grayscale or RGB intensity rate, motion illegibility (e.g., color and texture variations), inflexibility in the foreground or background segmentation, illumination variation, and low resolution which resist recognizing action accurately [9,10]. With the emergence of advanced technology, the redemption of accessible depth sensors is broadly used to achieve 3D action information. 3D information can be obtained through three approaches. The first approach is costly marker-based motion capture systems (MoCap) which uses visual sensing of markers settled in different parts of the human body and triangulation from several cameras to gain three-dimensional spatial information and the human skeleton. In the second approach, a stereo camera is used to acquire 3D depth information [11]. The stereo camera consists of two or more lenses with an individual image sensor or film frame for each lens. A stereo camera gives depth information by stereo matching and distance computation from lenses to object. The images captured by a stereo camera are sensitive to light changes and background clutter and action recognition from such images is a very challenging task [12]. The third approach involves the use of a depth sensor (for example Microsoft Kinect) that gives real-time 3D information for human body parts [11]. Unlike RGB camera, depth sensor camera gives overlapping multiple body portion information, it is insensitive to light changes that improve performance at dark, and in such data, it is easy to normalize the body orientation or its size variations [9]. This camera gives depth information from which skeleton data is obtained. The studies based on skeletal data often show high recognition performance, but where skeletal data is not available, the studies are not robust in terms of accuracy. These discussions encourage us to use depth information to establish an action recognition framework. However, DMMs based on the total depth frames of the entire video are not capable of obtaining the total motion information. To reduce this disability, in this paper, the depth map sequence of the entire video are partitioned into a set of overlapping portions. Each portion contains the same number of depth frames and DMMs sequences are constructed from DMMs of all portions. Then, the entire depth video is described through 3D auto-correlation features obtained from DMMs sequences. With the calculated features, the L_2 -regularized Collaborative Representation Classifier (L_2 -CRC) [13] and the Logarithmic Opinion Pool (LOGP) rule [14] work jointly to assign an action label of the video. The proposed framework is visualized in Figure 1.

Motivation and Contributions:

The method proposed by Chen et al. [15], used 3D auto-correlation features from depth map sequences for action recognition; however, their framework has limited performance with the same data. They did not achieve significant results through their framework. Therefore, the objective of our work is to develop a framework to increase the recognition results as well as the overall performance by using the 3D auto-correlation gradient features.

The main contributions of our work are listed below:

- The depth map sequences of each action video are partitioned into a set of sub-sequences of equal size. Afterward, DMMs are created from each sub-sequence corresponding to three projection views (front, side, and top) of 3D Euclidean space. Then, three DMMs sequences are derived by organizing all the DMMs along the projection views. The video is fragmented by two times generating two sets of sub-sequences using two different frame lengths and thus there are two sets of three DMMs sequences are obtained.
- Our recognition framework mines the 3D auto-correlation gradient feature vectors from three DMMs sequences by using the STACOG feature extractor instead of mining from depth map sequences as shown in [15].
- A decision fusion scheme is applied to combine residual outcomes obtained for two 3D action representation vectors.

- The proposed framework achieved the highest results as compared to all the other work done by applying the STACOG descriptor on depth video.

The remainder of this paper is organized as follows. A couple of action recognition frameworks are reviewed in Section 2. The proposed framework is described in Section 3. In Section 4, experimental results and discussion of the proposed framework are reported. Finally, the conclusion and future research directions are presented in Section 5.

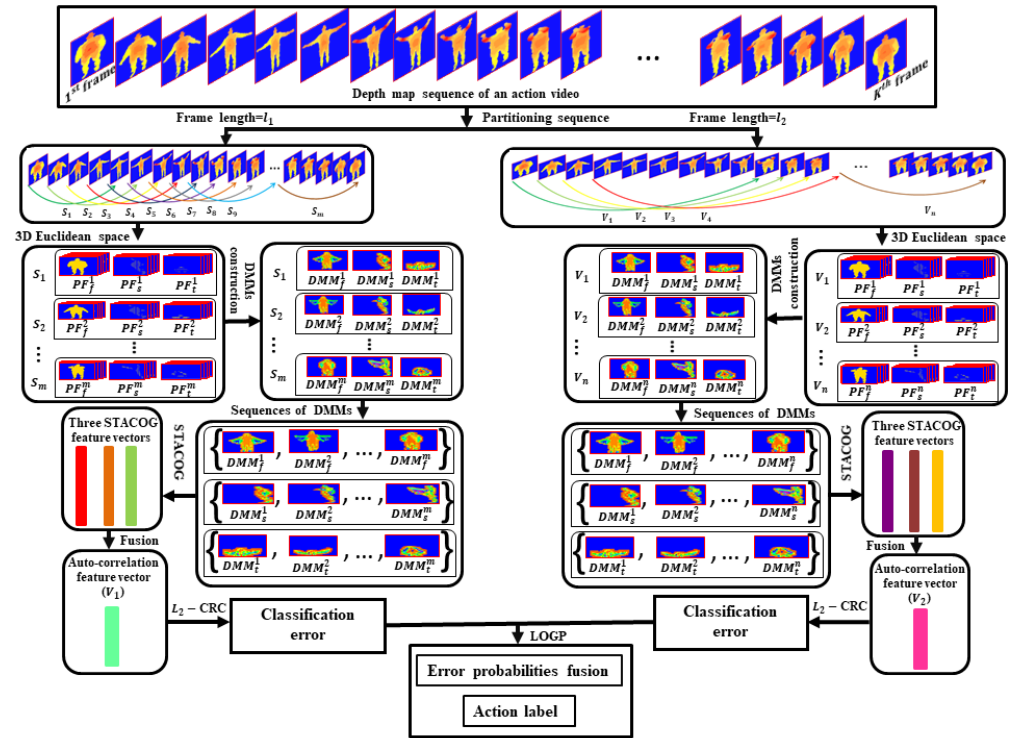


Figure 1. Proposed action recognition framework.

2. Related Work

This section describes current depth maps-based action recognition frameworks. Additionally, it also reviews skeleton, RGB, inertial, and fusion-based frameworks. Depending on depth data, Chen et al. [16] used local binary patterns (LBPs) to extract features. They represented two types of fusion levels and used the Kernel-based Extreme Learning Machine (KELM) for both levels. Ref. [17] introduced DMM-CT-HOG feature extractor that depends on Depth Motion Maps (DMMs), Contourlet Transform (CT), and Histogram of Oriented Gradients (HOGs). To improve accuracy, [18] used texture and dense shape information and combined them into DLE features that are fed to L_2 -regularized Collaborative Representation Classifier (L_2 -CRC). Ref. [19] proposed a method that fused classification results obtained by using multiple classifiers Kernel-based Extreme Learning Machine (KELM) through three types of features. A Bag-of-Map-Words (BoMW) method is introduced in [20] and feature vectors are extracted from Salient Depth Map (SDM) and Binary Shape Map (BSM) respectively and combined by the BoMW. Ref. [21] submitted a method using gradient local auto-correlations (GLAC) feature description algorithm based on spatial and orientational auto-correlations of local image. They introduced a fusion method depend on the Extreme Learning Machine classifier (ELM). Ji et al. [1], proposed a Spatio-Temporal Cuboid Pyramid (STCP) which subdivides the Depth Motion Sequence into spatial cuboids and temporal segments and used Histograms of Oriented Gradients (HOG) features. Chen et al. [22], used the texture feature descriptor Local Binary Pattern (LBP) and used the Kernel-based Extreme Learning Machine (KELM) classifier [19] to detect action. Again, in [23], DMMs are used as the feature descriptor. In their method, classification is accomplished by L_2 -CRC consisting of a distance-weighted Tikhonov ma-

trix. A new feature named Global Ternary Image (GTI) was introduced in [24]. By a bag of GTI model, the authors in [24] obtained data from motion regions and motion directions. After that, Liang et al. [25], used multiscale HOG descriptors and extracted local STACOG features. Then actions were recognized by L_2 -CRC classifier. To improve accuracy, [15] fused 2D and 3D auto-correlation of gradients features which are extracted by Gradient Local Auto-Correlations (GLAC) and STACOG descriptors, respectively. Then, the action is classified by KELM with RBF kernel. Liu et al. [26] presented a method that used Adaptive Hierarchical Depth Motion Maps (AH-DMMs) and Gabor filter. Their method can extract motion and shape cues without decreasing temporal information and adopt the Gabor filter to encode the texture data of AH-DMMs. Jin et al. [27] split depth maps into a set of sub-sequences to create a vague boundary sequence (VB-sequence). They obtained dynamic features by combining all DMMs of VB-sequences. After that, Zhang et al. [28], presented low-cost 3D histograms of texture feature descriptors by which discriminant features are obtained. They also introduced a multi-class boosting classifier (MBC) to use different features for recognition. Furthermore, Chen et al. [29] introduced a multi-temporal DMMs descriptor in which a non-linear weighting function is used to assemble depth frames. They used a patch-based Local Binary Pattern (LBP) feature descriptor to obtain texture information. They used Fisher kernel representation and used the KELM classifier [19] for action classification. Li et al. [30], extracted texture features by discriminative completed LBP (disCLBP) descriptor and used a hybrid classifier associated with Extreme Learning Machine (ELM) and collaborative representation classifier (CRC). The authors in [31] used Histogram of Oriented Gradients (HOG) and Pyramid Histogram of Oriented Gradients (PHOG) as shape feature descriptors. They used L_2 -CRC classifier. Azad et al. [32], introduced a multilevel temporal sampling (MTS) scheme that depended on the motion energy of depth maps. They extracted histograms of gradient and local binary patterns from a weighted depth motion map (WDMM). In [33], an action recognition scheme based on two types of depth images (generated using 3D Motion Trail Model (3DMTM)) was introduced. They obtained two features by using the GLAC algorithm from the images respectively and the features were fused in a vector. In the same year, Weiyao et al. [34] submitted Multilevel Frame Select Sampling (MFSS) model to obtain temporal samples from depth maps. They also proposed motion and static maps (MSM) and extracted texture features by the block-based LBP feature extraction scheme. They used the fisher kernel representation method to fuse obtained features and the KLM classifier to detect action. After that, Shekar et al. [35] introduced Stridden DMMs from which effective information of actions can be obtained quickly. They Undecimated the Dual-Tree Complex Wavelet Transform algorithm to extract wavelet (UDTCWT) features from the proposed DMMs. They used a Sequential Extreme Learning Machine classifier. To improve results, [36] used two types of images that are obtained by using the 3D Motion Trail Model (3DMTM). In their method feature vectors are mined from MHIs and SHIs by the GLAC feature descriptor. Al-Faris et al. [37] presented the construction of a multi-view region-adaptive multi-resolution-in-time depth motion map (MV-RAMDMM). They trained several scenes and time resolutions of the region-adaptive depth motion maps (RA-DMMs) by multi-stream 3D convolutional neural networks (CNNs). They used a multi-class SVMs classifier to recognize human actions.

Additionally, in [38], depth and inertial sensor-based features were extracted and fused to a single feature. The final feature set was passed to the collaborative representation classifier. Based on skeleton information, Youssef et al. [39], extracted normalized angles of local joints and used modified spherical harmonics (MSHs) to model the angular skeleton. They used MSH coefficients of the joints as the discriminative descriptor of the depth maps. Hou et al. [40], proposed a framework to convert Spatio-temporal data from skeleton sequence into color texture images. They used convolutional neural networks to obtain discriminative features. The authors in [41] created a Deep Convolutional Neural Network (3D2CNN) to acquire Spatio-temporal features from depth maps and calculated *JointVectors* from depth maps. The spatio-temporal features and *JointVectors* were passed individually

to the SVM classifier and the outputs were combined into a single result. To improve accuracy [42] introduced a Spatially Structured Dynamic Depth Images S^2 DDI to represent an action video. To generate S^2 DDI, they presented a non-scaling method and approved a multiply score fusion scheme to increase accuracy. Using RGB image, Al-Obaidi et al. [43] presented a method to anonymize action video. Histograms of oriented gradients (HOG) features are extracted from anonymized video images. A Generative Multi-View Action Recognition (GMVAR) method is presented in [44], by which three discrete scenarios are managed at the same time. They introduced a View Correlation Discovery Network (VCDN) to concatenate multi-view data. Liu et al. introduced dynamic pose images (DPI) and attention-based dynamic texture images (att-DTIs) in [45] to obtain spatial and temporal cues. They combined DPI and att-DTIs through multi-stream deep neural networks and a late fusion scheme. Inertial sensor-based low-level and high-level features are used in [46] to categorize human actions acted by a performer in real time. Haider et al. [47] introduced balanced, imbalanced, and super-bagging methods to recognize volleyball action. They used four wearable sensors to evaluate their method. Using signals created by the inertial measurement unit [48] introduced a method based on 1D-CNN construction and consider the tractability of features in time and duration. Bai et al. [49], presented a Collaborative Attention Mechanism (CAM) to develop Multi-view action recognition (MVAR) performance. They also proposed Mutual-Aid RNN (MAR) cell to obtain multi-view sequential information. Ullah et al. [50] introduced a conflux long short-term memory (LSTMs) network. They used CNN model to extract features and used SoftMax for classification. A fusion technique called View-Correlation Adaptation (VCA) in feature and label space was presented in [51]. They generated a semi-supervised feature augmentation (SeMix) and introduced a label-level fusion network. In [52], a light-weight CNN model was used to detect humans and LiteFlowNet CNN was proposed to extract features. The deep skip connection gated recurrent unit (DS-GRU) was used to recognize the action.

3. Proposed Recognition Framework

In this segment, we introduced the proposed framework with a detailed discussion on the construction of DMMs sequences, 3D auto-correlation features extraction, and action recognition. Algorithms 1 and 2 describe the mechanism of feature extraction and action recognition, respectively.

Algorithm 1 Algorithm for feature vector construction

Input: A Depth action video D of frame length L

Steps:

1. Split D and construct a set $\{S_j\}_{j=1}^m$, where $\text{len}(S_j) = l_1$ for all j
2. For all sub-sequences S_j , calculate DMM_f^j , DMM_s^j and DMM_t^j through Equation (1)
3. Use outcomes of **Step2** and generate $\{DMM_f^j\}_{j=1}^m$, $\{DMM_s^j\}_{j=1}^m$ and $\{DMM_t^j\}_{j=1}^m$
4. Use outcomes of **Step3** and calculate three feature vectors through Equations (3) and (4)
5. Concatenate outcomes of **Step4**
6. Further split D and construct another set $\{V_k\}_{k=1}^n$, where $\text{len}(V_k) = l_2$ for all k
7. Follow **Step2-Step5** for $\{V_k\}_{k=1}^n$

Output: Two auto-correlation feature vectors H_1 and H_2

Algorithm 2 Algorithm for action recognition

Input: The training feature set $Y = \{y_j\}_{j=1}^n$, test sample c , λ , K (number of action classes), class label k_i (for class partitioning), Q is the number of classifiers.

Steps:

1. Calculate $\hat{\gamma}_i$ using Equation (8)
2. **for** $Q \in \{1, 2\}$
 - for** $c \in \{H_1, H_2\} \leftarrow$ two feature vectors are calculated for c using Algorithm 1
 - for all** i **do**
 - Partition $Y_i, \hat{\gamma}_i$
 - Calculate $e_i = \|c - Y_i \hat{\gamma}_i\|_2$
 - Calculate $p_q(\omega|c)$ through Equation (11)
 - end for**
3. Calculate $P(\omega|c)$ through Equation (12)
4. Decide $class(c)$ through Equation (13)

Output: $class(c)$

3.1. Construction of DMMs Sequences

In our work, DMMs corresponding to three projection views (front, side, and top) are constructed for each sub-sequence of depth map sequence. To obtain DMMs, all the depth frames of each sub-sequence are projected onto 3D Euclidean space and projection frames corresponding to three projected views are generated. For each projected view, the addition of the utmost differences between sequential projection frames forms DMMs of front, side, and top.

To interpret computation of DMMs sequence [23], at first, a depth video D of length L is divided into a set $\{S_j\}_{j=1}^m$ of sub-sequences of uniform size $l_1 > 0$ as $D = \cup_{j=1}^m S_j$, where j represents the index of sub-sequence. Let us consider a depth frame sequence $\{p^1, p^2, p^3, \dots, p^{l_1}\}$ for each sub-sequence, where l_1 is the frame length of each sub-sequence, i.e., $len(S_j) = l_1$ for all j . The projection of i th frame p^i on 3D Euclidean space provides three projected frames p_v^i (which are referred to as PF_v^i in Figure 1), where v designates front, side, and top projection views and $v \in \{f, s, t\}$. The DMMs corresponding to projection views are defined by the following equation:

$$DMM_v = \sum_{i=1}^{l_1-1} |p_v^{i+1} - p_v^i|, \quad (1)$$

For all S_j , DMMs are represented by DMM_f^j , DMM_s^j , and DMM_t^j . Therefore, $\{DMM_f^1, DMM_f^2, \dots, DMM_f^m\}$, $\{DMM_s^1, DMM_s^2, \dots, DMM_s^m\}$ and $\{DMM_t^1, DMM_t^2, \dots, DMM_t^m\}$ sequences are formed from $\{S_j\}_{j=1}^m$ of D . In action datasets, the same actions are performed by different individuals with different speeds. To cope with action speed variations, the depth map sequence of D is further divided into another set $\{V_k\}_{k=1}^n$ of sub-sequences where frame length of each sub-sequence is $l_2 > 0$, i.e., $len(V_k) = l_2$ for all k (see Figure 1). As a result, more three new sets of DMMs sequences $\{DMM_f^1, DMM_f^2, \dots, DMM_f^n\}$, $\{DMM_s^1, DMM_s^2, \dots, DMM_s^n\}$ and $\{DMM_t^1, DMM_t^2, \dots, DMM_t^n\}$ are obtained from $\{V_k\}_{k=1}^n$. In our DMMs sequences constructing mechanism, numerical values of frame lengths l_1 and l_2 are experimentally chosen to 5 and 10 respectively. The frame length of a sub-sequence may vary and must be set to less than the length of total depth video, i.e., l_1 and $l_2 < L$. The DMMs sequences constructing scheme for frames length l_1 is displayed in Figure 2. The frame interval I in Figure 2 is set to 1 which is the number of frames from the first frame of a portion to the first frame of the neighboring portion which indicates how many

frames between the two portions are overlapped. Please note that the frame interval must be less than the frame length of a sub-sequence, i.e., $I < l_1$ and l_2 .

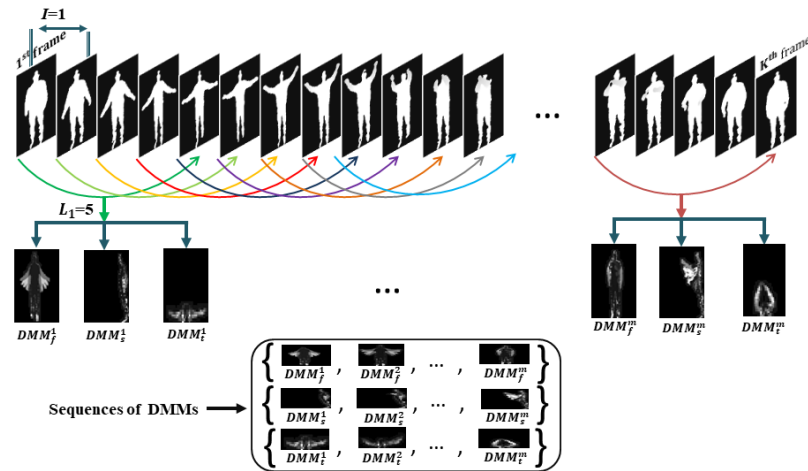


Figure 2. Construction of DMMs sequences according to sub-sequences of 5 frames.

3.2. Action Vector Formation

STACOG was introduced in [53] for RGB video sequences to extract local relationships within the space-time gradients of three-dimensional motion by using auto-correlation functions to space-time orientations and the magnitudes of the gradients. In our work, this method is applied to all the DMMs sequences (calculated in the previous section) of a depth video D to extract 3D geometric features of human motion. At each space-time volume $S(x, y, t)$ (in general, this volume stands for a DMMs sequence) around each space-time point in a DMMs sequence the space-time gradient vector is computed through the derivatives S_x , S_y , and S_t to extract features. The space-time gradients can be described by angles $\alpha = \arctan(S_x, S_y)$ and $\beta = \arcsin(\frac{S_t}{m_g})$, where the magnitude of gradient is defined by $m_g = \sqrt{(S_x^2 + S_y^2 + S_t^2)}$. By the two angles, space-time orientation of the gradient is coded into B orientation bins on a unit sphere by selecting weights to the nearest bins (see Figure 3). Finally, the orientation is represented by B -dimensional vector named space-time orientation coding (STOC) vector which is denoted by \mathbf{b} . By using the magnitude m_g and the STOC vector \mathbf{b} of the gradients, the N th order auto-correlation function for the space-time gradients is defined as follows:

$$\mathbf{R}_N(\mathbf{d}_1, \dots, \mathbf{d}_N) = \int f[m_g(\mathbf{p}), \dots, m_g(\mathbf{p} + \mathbf{d}_N)] \mathbf{b}(\mathbf{p}) \otimes \dots \otimes \mathbf{b}(\mathbf{p} + \mathbf{d}_N) d\mathbf{p} \quad (2)$$

where $\mathbf{d}_i = (\mathbf{d}_1, \dots, \mathbf{d}_N)$ displacement are vectors from the reference point $\mathbf{p} = (x, y, t)$, f represents a weighting function and \otimes is the tensor product of vector. In the tensor products, there are small numbers of non-zero components related to the gradient orientations of the neighboring vectors. The parameters $N \in \{0, 1\}$; $d_{1x,y} \in \{\pm \Delta s, 0\}$; $d_{1t} \in \{\Delta t, 0\}$; $f(\cdot) \equiv \min(\cdot)$ are confined in the experiment. Where Δs is the displacement interval along the spatial axis and Δt is that of along the temporal axis. To inhibit the effect of isolated noise on surrounding auto-correlations, min is received regarding to weight function f .

For $N \in \{0, 1\}$ the 0th order and the 1st order STACOG features can be written as,

$$S_0 = \sum_{\mathbf{p}} m_g(\mathbf{p}) \mathbf{b}(\mathbf{p}), \quad (3)$$

$$S_1(\mathbf{d}_1) = \sum_{\mathbf{p}} \min[m_g(\mathbf{p}), m_g(\mathbf{p} + \mathbf{d}_1)] \mathbf{b}(\mathbf{p}) \mathbf{b}(\mathbf{p} + \mathbf{d}_1)^T, \quad (4)$$

where S_0 and S_1 are 0th and 1st order auto-correlations which gives the 0th order and the 1st order STACOG features, and T is the transpose.

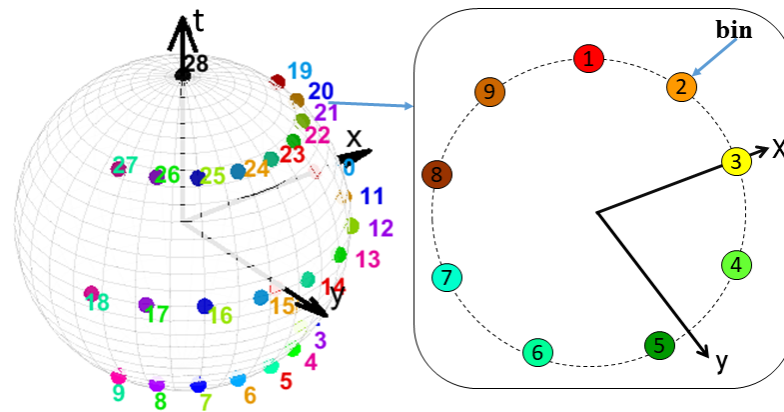


Figure 3. 28 orientation bins along latitude and longitude on a hemisphere. 4 orientation bin layers including a layer at pole are in two-dimensional $x - y$ plane. 9 orientation bins are located on each layer except at the pole (contains one bin).

3.3. Action Recognition

By applying Algorithm 1, two auto-correlation feature vectors H_1 and H_2 are acquired corresponding to two different sets of sub-sequences, $\{S_j\}_{j=1}^m$ and $\{V_k\}_{k=1}^n$, of the depth video D (see Figure 1). The dimension of H_1 and H_2 are reduced through Principal Component Analysis (PCA) [54]. Then the two vectors are passed separately to L_2 -regularized Collaborative Representation Classifier (L_2 -CRC) [13] and the relevant two distinct outcomes are fused by logarithmic opinion pool (LOGP) [14]. To explain L_2 -CRC, let us denote the class number by K . The set $Y = [Y_1, Y_2, Y_3, \dots, Y_i, \dots, Y_K] = [y_1, y_2, y_3, \dots, y_j, \dots, y_n] \in R^{(d \times n)}$ is the set of all training samples, where d is the dimensionality of training samples, m is the number of training samples from K classes, $Y_i \in R^{(d \times m_i)}$ is subset of training samples from class i and $y_j \in R^d$ is any training sample of Y_i . Let, $c \in R^d$ be any unknown test sample which is defined by the linear combination of all the training samples in Y :

$$c \approx Y\gamma, \quad (5)$$

where $\gamma = [\gamma_1, \gamma_2, \gamma_3, \dots, \gamma_i, \dots, \gamma_K]$ is a $m \times 1$ coefficients vector associated with the training samples of class i . In practice, Equation(5) cannot be solved directly because it is under determination [55]. By the solution of the following norm minimization problem, Equation(5) can be solved:

$$\arg \min_{\gamma} \{ \|c - Y\gamma\|_2^2 + \lambda \|M\gamma\|_2^2 \}, \quad (6)$$

subject to $c \approx Y\gamma,$

where λ denotes the regularization parameter and M is the Tikhonov regularization matrix [56], which is configured by the following diagonal matrix.

$$M = \begin{bmatrix} \|c - y_1\|_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \|c - y_n\|_2 \end{bmatrix}, \quad (7)$$

The coefficient vector can be calculated as [57],

$$\hat{\gamma} = (Y^T Y + \lambda M^T M)^{-1} Y^T c = Zc, \quad (8)$$

Since the training samples are Y is given and λ is determined by these samples then Z can be simply calculated and thus Z is independent of c . It is clear when the test sample c is given, the corresponding vector $\hat{\gamma}$ can be easily computed from Equation (8). The coefficient

vector $\hat{\gamma}$ is represented as $[\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3, \dots, \hat{\gamma}_i, \dots, \hat{\gamma}_K]$ by considering all the action classes. Now, the class-specific residual error can be obtained by

$$e_i = \|c - Y_i \hat{\gamma}_i\|_2, \quad (9)$$

where, Y_i is the dictionary sample and $\hat{\gamma}_i$ is the coefficient of i th class, respectively.

From Equation (9), an error vector is obtained about an input feature vector. In our case, there are two error vectors $e^1 = [e_1^1, e_2^1, e_3^1, \dots, e_i^1, \dots, e_K^1]$ and $e^2 = [e_1^2, e_2^2, e_3^2, \dots, e_i^2, \dots, e_K^2]$ since we input two feature vectors H_1 and H_2 obtained by Algorithm 1 for the test sample c . A decision fusion scheme logarithmic opinion pool (LOGP) [14] rule is used to concatenate the probabilities of those errors and to output the class label. In this scheme, the following global membership function is calculated through the posterior probability $p_q(\omega|c)$ of each classifier.

$$P(\omega|c) = \prod_{q=1}^Q p_q(\omega|c)^{\frac{1}{Q}}, \quad (10)$$

where $\omega \in [1, 2, 3, \dots, i, \dots, K]$ denotes class label, and $Q (= 2)$ denotes the number of classifiers.

Then a Gaussian mass function corresponding to the residual error $e = [e_1, e_2, e_3, \dots, e_i, \dots, e_K]$ is represented by the following equation.

$$p_q(\omega|c) \approx \exp(-e_i), \quad (11)$$

Equation (11) defines the higher posterior probability $p_q(\omega|c)$ for a smaller residual error e_i . Therefore, the combined probability from the two classifiers is defined as:

$$P(\omega|c) = \exp(-e_i^1)^{\frac{1}{2}} \times \exp(-e_i^2)^{\frac{1}{2}}, \quad (12)$$

$$\text{And, } \text{class}(c) = \max\{P(\omega|c)\}, \quad (13)$$

where e^1 and e^2 are normalized to $[0, 1]$.

4. Experimental Results and Discussion

This section discusses three sets of experiments on three datasets to evaluate the performance of the proposed framework. First, the datasets are introduced along with their challenges. Secondly, the setup of STACOG parameters is then discussed to evaluate the proposed framework. Finally, experimental results on three datasets are described.

4.1. Datasets

Our proposed framework is greatly appraised on depth-based actions datasets named MSR-action 3D dataset [58], DHA dataset [59], and UTD-MHAD dataset [38].

4.1.1. MSR-Action 3D Dataset

MSR-Action 3D dataset is captured by a depth camera which represents action data of depth map sequences. The resolution of each map is 320×240 . This dataset has 20 types of action categories. All the actions are acted by 10 different persons and every subject act in each action 2 or 3 times. In this dataset, the number of depth map sequences is 557 [58]. This dataset is a challenging because of the correspondence between some actions (e.g., “Draw x” and “Draw tick”).

4.1.2. DHA Dataset

DHA dataset was introduced in [59] which contains some actions extended from the Weizmann dataset [60]. The Weizmann dataset is used in action recognition based on RGB sequences. The DHA dataset involves 23 action types among which 1 to 10 actions are adopted from Weizmann dataset [61]. All the actions are performed by 21 subjects (12 males and nine females) and the total number of depth map sequences is 483. Because of the

inter-similarity between action classes (e.g., “rod-swing” and “golf swing”), the DHA dataset is challenging.

4.1.3. UTD-MHAD Dataset

In the UTD-MHAD dataset [38], RGB videos, depth videos, skeleton positions, and inertial signals are captured by a video sensor and a wearable inertial sensor. All the actions of this dataset contain 27 actions and all the actions are performed by eight subjects (four females and four males). Each performer repeats each action four times. This dataset includes 861 depth action sequences, after eliminating three inappropriate sequences.

4.2. Parameter Setting

The proposed framework is evaluated on the datasets discussed above and compared with the other state-of-the-art approaches. Of all the samples of each dataset, some samples are used as training samples and the remaining samples are used as test samples. Depending on the test samples, results on all datasets are obtained. Each depth action video of all datasets is partitioned into sub-sequences using the same frame lengths. The frame interval between two consecutive sub-sequences is set to 1 which indicates the number of overlapping frames. Thus, for two different frame lengths 5 and 10, the overlapping frames 4 and 9 are obtained, respectively. Additionally, for all action datasets, we used the same values of parameters. At first, all parameter values are tuned for a dataset to query which values give the highest recognition accuracy. Then, the values of parameters set for the highest result are used in all other datasets to verify the superiority of the framework. To extract STACOG features, orientation bins in the $x - y$ plane and orientation bin layers are set to 9 and 4, respectively. According to [15], the temporal interval is set to 1 and the spatial interval is fixed to 8. The L_2 -CRC parameter λ is tuned to 0.0001.

4.3. Classification on MSR-Action 3D Dataset

In the experimental arrangement, we used all action categories of MSR-Action 3D dataset instead of dividing them into different action subsets. The action samples acted through persons of the odd number are employed as training samples (284) and the samples of the remaining persons of even number are used as test samples (273). Our proposed framework gives 93.4% recognition accuracy which is compared with other frameworks on depth data as shown in Table 1. Among 20 actions, the classification accuracy is 100% for 14 actions. The remaining 6 actions have some confusion with other actions because of some inter-class similarities. For example, the confusion of an action “Side kick” with an action “Hand catch” is 9.1% (see Figure 4). The accuracy including confusion information of each class is further clarified in Table 2.

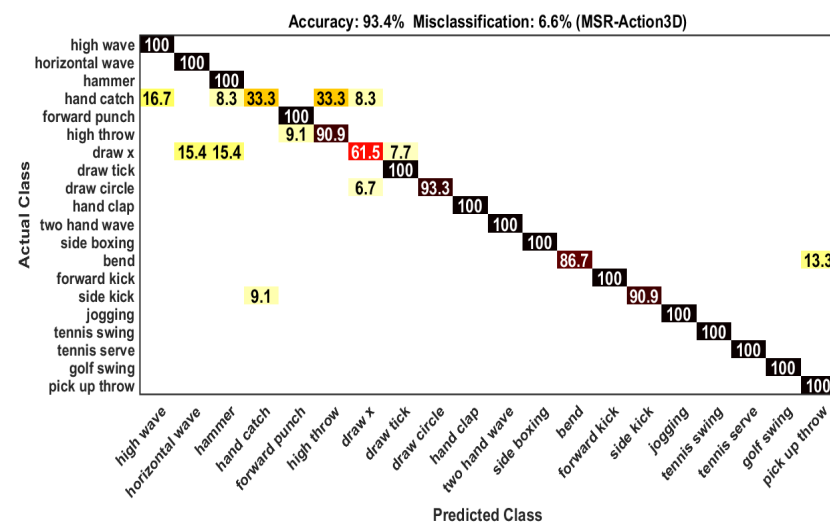


Figure 4. Confusion matrix on MSR-Action 3D dataset.

Table 1. Comparison of action recognition accuracy (%) with state-of-the-art frameworks on the MSR-Action 3D dataset.

Approach	Accuracy (%)
Decision-level Fusion (MV) [19]	91.9
DMM-GLAC-FF [16]	89.38
DMM-GLAC-DF [16]	92.31
DMM-LBP-FF [21]	91.9
DMM-LBP-DF [21]	93.0
3D ² CNN [41]	84.07
Skeleton-MSH [39]	90.98
3DHoT_S [28]	91.9
3DHoT_M [28]	88.3
Depth-STACOG [15]	75.82
DMM-GLAC [15]	89.38
WDMM [32]	90.0
DMM-UDTCWT [35]	92.67
Proposed Approach	93.4

Table 2. Class-specific accuracy on MSR-Action3D dataset.

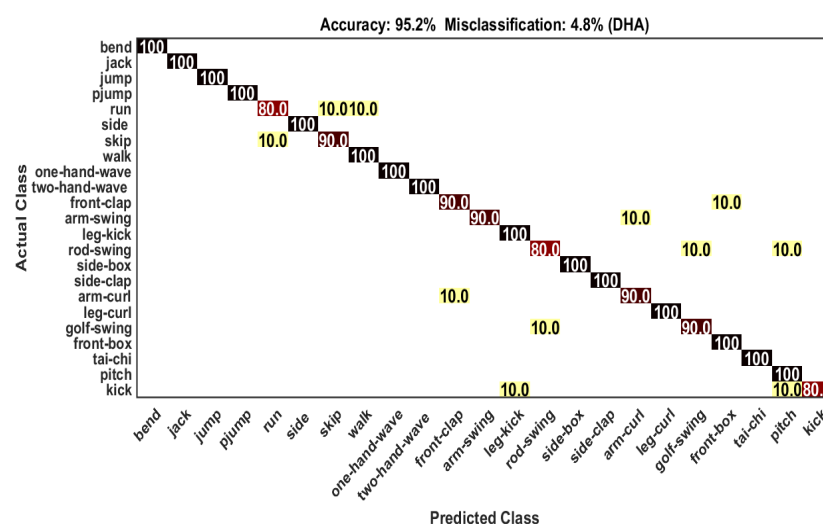
Actions	Classification (%)	Confusion (%)
High wave	100	No confusion
Horizontal wave	91.7	Hammer (8.3)
Hammer	100	No confusion
Hand catch	33.3	High wave (16.7), Hammer (8.3), High throw (33.3), Draw x (8.3)
Forward punch	100	No confusion
High throw	90.9	Forward punch (9.1)
Draw x	61.5	Horizontal wave (15.4), Hammer (15.4), Draw tick (7.7)
Draw tick	100	No confusion
Draw circle	93.3	Draw X (6.7)
Hand clap	100	No confusion
Two hand wave	100	No confusion
Side boxing	100	No confusion
Bend	86.7	Pick up and throw (13.3)
Forward kick	100	No confusion
Side kick	90.9	Hand catch (9.1)
Jogging	100	No confusion
Tennis swing	100	No confusion
Tennis serve	100	No confusion
Golf swing	100	No confusion
Pick up and throw	100	No confusion

4.4. Classification on DHA Dataset

In the DHA dataset, samples of the odd subjects are used as training samples and the samples of the even subjects are used as test samples. There are 253 samples are used as training samples and 230 samples are used as test samples. Our proposed framework achieves 95.2% accuracy which shows the effectiveness of the recognition framework. From Table 3, we can observe that 15 out of 23 actions are recognized with 100% accuracy. The remaining 8 actions are confused with other actions shown in Figure 5. The action “golf swing” gives 10% confusion with “rod-swing”. The comparison of our recognition framework with other state-of-the-art methods is shown in Table 3. It is clear from the table that our proposed framework beats other existing frameworks considerably. The class-wise classification accuracy (for right and wrong classification) is shown in Table 4.

Table 3. Comparison of action recognition accuracy (%) with state-of-the-art frameworks on the DHA dataset.

Approach	Accuracy (%)
SDM-BSM [20]	89.50
GTI-BoVW [24]	91.92
Depth WDM [32]	81.05
RGB-VCDN [44]	84.32
VCDN [44]	88.72
Binary Silhouette [43]	91.97
DMM-UDTCWT [35]	94.2
Stridden DMM-UDTCWT [35]	94.6
VCA [51]	89.31
CAM [49]	87.24
Proposed Approach	95.2

**Figure 5.** Confusion matrix on DHA dataset.**Table 4.** Class-specific accuracy on DHA dataset.

Actions	Classification (%)	Confusion (%)
Bend	100	No confusion
Jack	100	No confusion
Jump	100	No confusion
Pjump	100	No confusion
Run	80.0	Skip (10.0), Walk (10.0)
Side	100	No confusion
Skip	90.0	Run (10.0)
Walk	100	No confusion
One-hand-wave	100	No confusion
Two-hand-wave	100	No confusion
Front-clap	90.0	Front-box (10.0)
Arm-swing	90.0	Arm-curl (10.0)
Leg-kick	100	No confusion
Rod-swing	80.0	Golf-swing (10.0), Pitch (10.0)
Side-box	100	No confusion
Side-clap	90.0	Side-box (10.0)
Arm-curl	90.0	Front-clap (10.0)
Leg-curl	100	No confusion
Golf-swing	90.0	Rod-swing (10.0)
Front-box	100	No confusion
Tai-chi	100	No confusion
Pitch	100	No confusion
Kick	90.0	Pitch (10.0)

4.5. Classification on UTD-MHAD Dataset

In the UTD-MHAD dataset, samples of the odd subjects are used as training samples (431) and the samples of the even subjects are used as test samples (430). The evaluation result of our framework on this dataset gives 87.7% recognition accuracy (see Table 5) because of using varieties actions. The result in our recognition framework gives 100% accuracy for 11 actions and the remaining 16 actions show confusion with other actions (see Figure 6). The individual class recognition performance is reported in Table 6.

Table 5. Comparison of action recognition accuracy (%) with state-of-the-art frameworks on the UTD-MHAD dataset.

Approach	Accuracy (%)
Kinect [38]	66.10
Inertial [38]	67.20
Kinect+Inertial [38]	79.10
DMM-EOH [19]	75.3
DMM-LBP [19]	84.20
CNN-Top [40]	74.65
CNN-Fusion [40]	86.97
3DHOT-MBC [28]	84.40
VDDMMs [27]	85.10
Structured body DDI [42]	66.05
Structured part DDI [42]	78.70
RGB DTIs [45]	85.39
Inertial [48]	85.35
Proposed Approach	87.7

Table 6. Class-specific accuracy on UTD-MHAD dataset.

Actions	Classification (%)	Confusion (%)
Swipe-lift	87.5	Arm-cross (12.5)
Swipe-right	100	No confusion
Wave	81.3	Swipe-right (12.5), Draw-circle-CW (6.3)
Clap	43.8	Arm-cross (31.3), Arm-curl (25.0)
Throw	87.5	Draw-circle (CCW) (6.3), Tennis-serve (6.3)
Arm-cross	81.3	Arm-curl (18.8)
Basketball-shoot	81.3	Arm-curl (6.3), Tennis-serve (12.5)
Draw-x	100	No confusion
Draw-circle CW	93.8	Catch (6.3)
Draw-circle (CCW)	81.3	Draw X (6.3), Arm-curl (15.5)
Draw-triangle	43.8	Draw-circle (CCW) (56.3)
Bowling	100	No confusion
Boxing	100	No confusion
Baseball-swing	100	No confusion
Tennis-swing	81.3	Bowling (12.5), Baseball-swing (6.3)
Arm-curl	75.0	Arm-cross (12.5), Basketball-shoot (6.3), Push (6.3)
Tennis-serve	100	No confusion
Push	87.5	Arm-curl (12.5)
Knock	81.3	Arm-cross (6.3), Arm-curl (6.3), Catch (6.3)
Catch	75.0	Draw-triangle (18.8), Knock (6.3)
Pickup-throw	93.8	Tennis-serve (6.3)
Jog	100	No confusion
Walk	100	No confusion
Sit2stand	100	No confusion
Stand2sit	100	No confusion
Lunge	93.8	Bowling (6.3)
Squat	100	No confusion

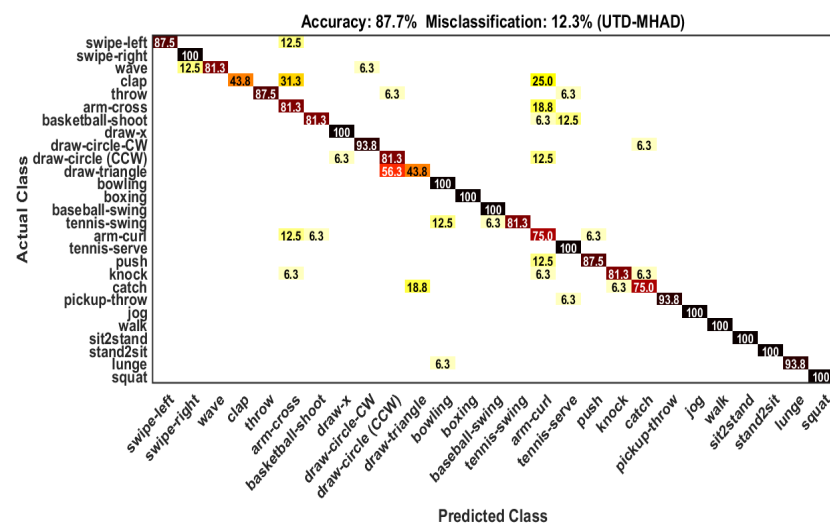


Figure 6. Confusion matrix on the UTD-MHAD dataset.

4.6. Efficiency Evaluation

The execution time and the space complexity of key factors are deliberated to show the efficiency of our system.

4.6.1. Execution Time

The system is executed by using MATLAB on CPU platform with an Intel i5-7500 Quad-core processor of 3.41 GHz frequency and a RAM of 16 GB. There are seven major components in the proposed approach: DMMs sequences construction for frame length 5, DMMs sequences construction for frame length 10, H_1 feature vector generation, H_2 feature vector generation, PCA on H_1 , PCA on H_2 , Action label. The execution time of these components is determined to assess the time efficiency of the system on three datasets as MSR-Action 3D, DHA, and UTD-MHAD dataset. Table 7 showed the execution time (in milliseconds) of the seven components on those datasets and compared the total execution time on the datasets. In the case of the MSR-Action 3D dataset, execution times are calculated for each action sample with 40 frames on average. As can be seen from Table 7, 40 frames are processed in less than one second (i.e., 252.6 ± 74.8 milliseconds). Therefore, our proposed recognition framework can be used for real-time action recognition on the MSR-Action 3D dataset. The execution times on the DHA dataset are calculated for each action sample with 29 frames on average. Table 7 showed that the 29 frames are processed in less than one second (i.e., 379.1 ± 90.7 milliseconds) which proves our framework can be used for real-time action recognition on the DHA dataset. Table 7 also presented the execution times (in milliseconds) on the UTD-MHAD dataset for each action sample with 68 frames on average. To process 68 frames, the system requires less than one second (i.e., 508.9 ± 100.9 milliseconds) which showed the capability of the real-time action recognition of our proposed framework.

4.6.2. Space Complexity

The components PCA and L_2 -CRC are the key components for the calculation of space complexity of the proposed system. PCA and L_2 -CRC are adopted for both frame lengths 5 and 10. Therefore, the complexity of PCA is $2 * O(l^3 + l^2m)$ [23] and the complexity of L_2 -CRC is $2 * O(n_c \times m)$ [62]. Thus, the total complexity of the system can be expressed as $2 * O(l^3 + l^2m) + 2 * O(n_c \times m)$. Table 8 describes the computed complexity and compared it with the complexities of other existing frameworks. The table shows that our framework delivers lower complexity and recognizes actions better than other existing frameworks.

Table 7. Comparison of execution time (mean \pm std) of the key factors on three datasets.

Main Components	MSR-Action3D Dataset	DHA Dataset	UTD-MHAD Dataset
DMMs sequences construction for frame length 5	11.3 \pm 0.7	80.7 \pm 6.2	36.6 \pm 3.0
DMMs sequences construction for frame length 10	18.8 \pm 1.3	155.6 \pm 12.0	67.9 \pm 5.5
H_1 feature vector generation	108.5 \pm 35.0	69.9 \pm 35.0	197.7 \pm 45.6
H_2 feature vector generation	95.6 \pm 36.7	57.2 \pm 36.7	180.9 \pm 45.9
PCA on H_1	8.5 \pm 0.4	7.3 \pm 0.3	10.7 \pm 0.3
PCA on H_2	8.4 \pm 0.3	7.2 \pm 0.3	10.7 \pm 0.3
Action label	1.5 \pm 0.4	1.2 \pm 0.2	4.4 \pm 0.3
Total execution time	252.6 \pm 74.8/action sample (40 frames)	379.1 \pm 90.7/action sample (29 frames)	508.9 \pm 100.9/action sample (68 frames)

Table 8. Comparison of computational complexity of the proposed approach with other existing approaches.

Approach	Components	Space Complexity
DMM [23]	PCA, L_2 -CRC	$O(l^3 + l^2m) + O(n_c \times m)$ l = size of action vector, m = number of training samples, n_c = number of action classes
DMM-LBP-DF [21]	PCA, Kernel-based Extreme Learning Machine (KELM)	$O(l^3 + l^2m) + 3 * O(m^3)$ l = size of action vector, m = number of training samples
MHF+SHF+KELM [13]	PCA, KELM	$O(l^3 + l^2m) + 2 * O(m^3)$ l = size of action vector, m = number of training samples
GMSHI+GSHI+CRC [36]	PCA, L_2 -CRC	$O(l^3 + l^2m) + O(n_c \times m)$ l = size of action vector, t = number of training samples, n_c = number of action classes
Enhanced auto-correlation [63]	PCA, KELM ensemble	$O(l^3 + l^2m) + O(m^3)$ l = size of action vector, m = number of action classes
Proposed Approach	PCA, L_2 -CRC	$2 * O(l^3 + l^2m) + 2 * O(n_c \times m)$ l = size of action vector, m = number of training samples, n_c = number of action classes

5. Conclusions

In this paper, we present an effective action recognition framework that is based on 3D Auto-Correlation features. In fact, the Depth Motion Maps (DMMs) sequence representation is firstly introduced to obtain additional temporal motion information from depth map sequences which can distinguish similar actions. The space-time auto-correlation of gradients features description algorithm is then used to extract motion cues from the sequences of DMMs according to different projection views. At last, the Collaborative representation classifier (CRC) and the decision fusion scheme are used for detecting action class. Experimental results on three benchmark datasets shows that the proposed framework is better than the state-of-the-art methods. Moreover, the framework outperforms other existing techniques that are based on space-time auto-correlation of gradients feature. Furthermore, the space-time complexity analysis of the proposed framework indicates that it can be used for the real-time human action recognition.

Author Contributions: M.F.B. : Conceptualization, methodology, software, data curation, validation, formal analysis, investigation, writing—original draft preparation; S.T.: Validation, Implementation, formal analysis, writing—original draft preparation; H.A.: Conceptualization, writing—original draft preparation, writing—review and editing; W.Z.: Writing—review and editing, funding acquisition, data collection and analysis; M.Y.L.: Methodology, formal analysis, revision—draft preparation and critical revision, funding acquisition, project administration; A.U.: investigation, visualization, writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript, please turn to the <http://img.mdpi.org/data/contributor-role-instruction.pdf> CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07043302).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ji, X.; Cheng, J.; Feng, W. Spatio-temporal cuboid pyramid for action recognition using depth motion sequences. In Proceedings of the 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 14–16 February 2016; pp. 208–213.
2. Li, R.; Liu, Z.; Tan, J. Exploring 3D human action recognition: From offline to online. *Sensors* **2018**, *18*, 633.
3. Fan, Y.; Weng, S.; Zhang, Y.; Shi, B.; Zhang, Y. Context-Aware Cross-Attention for Skeleton-Based Human Action Recognition. *IEEE Access* **2020**, *8*, 15280–15290.
4. Cho, S.; Maqbool, M.H.; Liu, F.; Foroosh, H. Self-Attention Network for Skeleton-based Human Action Recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CA, USA, 1–5 March 2020; pp. 624–633.
5. Ali, H.H.; Moftah, H.M.; Youssif, A.A. Depth-based human activity recognition: A comparative perspective study on feature extraction. *Future Comput. Inform. J.* **2018**, *3*, 51–67.
6. Tufek, N.; Yalcin, M.; Altintas, M.; Kalaoglu, F.; Li, Y.; Bahadir, S.K. Human action recognition using deep learning methods on limited sensory data. *IEEE Sens. J.* **2019**, *20*, 3101–3112.
7. Elbasiony, R.; Gomaa, W. A survey on human activity recognition based on temporal signals of portable inertial sensors. In *International Conference on Advanced Machine Learning Technologies and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 734–745.
8. Masum, A.K.M.; Bahadur, E.H.; Shan-A-Alahi, A.; Chowdhury, M.A.U.Z.; Uddin, M.R.; Al Noman, A. Human Activity Recognition Using Accelerometer, Gyroscope and Magnetometer Sensors: Deep Neural Network Approaches. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
9. Jalal, A.; Kim, Y.H.; Kim, Y.J.; Kamal, S.; Kim, D. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recognit.* **2017**, *61*, 295–308.
10. Farooq, A.; Won, C.S. A survey of human action recognition approaches that use an RGB-D sensor. *IEIE Trans. Smart Process. Comput.* **2015**, *4*, 281–290.

11. Chen, C.; Jafari, R.; Kehtarnavaz, N. A survey of depth and inertial sensor fusion for human action recognition. *Multimed. Tools Appl.* **2017**, *76*, 4405–4425.
12. Aggarwal, J.K.; Xia, L. Human activity recognition from 3d data: A review. *Pattern Recognit. Lett.* **2014**, *48*, 70–80.
13. Bulbul, M.F.; Islam, S.; Zhou, Y.; Ali, H. Improving Human Action Recognition Using Hierarchical Features And Multiple Classifier Ensembles. *Comput. J.* **2019**, doi:10.1093/comjnl/bxz123.
14. Benediktsson, J.A.; Sveinsson, J.R. Multisource remote sensing data classification based on consensus and pruning. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 932–936.
15. Chen, C.; Zhang, B.; Hou, Z.; Jiang, J.; Liu, M.; Yang, Y. Action recognition from depth sequences using weighted fusion of 2D and 3D auto-correlation of gradients features. *Multimed. Tools Appl.* **2017**, *76*, 4651–4669.
16. Chen, C.; Hou, Z.; Zhang, B.; Jiang, J.; Yang, Y. Gradient local auto-correlations and extreme learning machine for depth-based activity recognition. In *International Symposium on Visual Computing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 613–623.
17. Bulbul, M.F.; Jiang, Y.; Ma, J. Human action recognition based on DMMs, HOGs and Contourlet transform. In Proceedings of the 2015 IEEE International Conference on Multimedia Big Data, Beijing, China, 20–22 April 2015; pp. 389–394.
18. Bulbul, M.F.; Jiang, Y.; Ma, J. Real-time human action recognition using DMMs-based LBP and EOH features. In *International Conference on Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 271–282.
19. Bulbul, M.F.; Jiang, Y.; Ma, J. DMMs-based multiple features fusion for human action recognition. *Int. J. Multimed. Data Eng. Manag.* **2015**, *6*, 23–39.
20. Liu, H.; Tian, L.; Liu, M.; Tang, H. Sdm-bsm: A fusing depth scheme for human action recognition. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 4674–4678.
21. Chen, C.; Jafari, R.; Kehtarnavaz, N. Action recognition from depth sequences using depth motion maps-based local binary patterns. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015; pp. 1092–1099.
22. Chen, C.; Liu, M.; Zhang, B.; Han, J.; Jiang, J.; Liu, H. 3D Action Recognition Using Multi-Temporal Depth Motion Maps and Fisher Vector. In Proceedings of the International Joint Conference On Artificial Intelligence IJCAI, New York, NY, USA, 9–15 July 2016; pp. 3331–3337.
23. Chen, C.; Liu, K.; Kehtarnavaz, N. Real-time human action recognition based on depth motion maps. *J. Real-Time Image Process.* **2016**, *12*, 155–163.
24. Liu, M.; Liu, H.; Chen, C.; Najafian, M. Energy-based global ternary image for action recognition using sole depth sequences. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 47–55.
25. Liang, C.; Qi, L.; Chen, E.; Guan, L. Depth-based action recognition using multiscale sub-actions depth motion maps and local auto-correlation of space-time gradients. In Proceedings of the 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), Niagara Falls, NY, USA, 6–9 September 2016; pp. 1–7.
26. Liu, H.; He, Q.; Liu, M. Human action recognition using adaptive hierarchical depth motion maps and Gabor filter. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 1432–1436.
27. Jin, K.; Jiang, M.; Kong, J.; Huo, H.; Wang, X. Action recognition using vague division DMMs. *J. Eng.* **2017**, *2017*, 77–84.
28. Zhang, B.; Yang, Y.; Chen, C.; Yang, L.; Han, J.; Shao, L. Action recognition using 3D histograms of texture and a multi-class boosting classifier. *IEEE Trans. Image Process.* **2017**, *26*, 4648–4660.
29. Chen, C.; Liu, M.; Liu, H.; Zhang, B.; Han, J.; Kehtarnavaz, N. Multi-temporal depth motion maps-based local binary patterns for 3-D human action recognition. *IEEE Access* **2017**, *5*, 22590–22604.
30. Li, W.; Wang, Q.; Wang, Y. Action Recognition Based on Depth Motion Map and Hybrid Classifier. *Math. Probl. Eng.* **2018**, doi:10.1155/2018/8780105.
31. Bulbul, M.F. Searching Human Action Recognition Accuracy from Depth Video Sequences Using HOG and PHOG Shape Features. *Eur. J. Appl. Sci.* **2018**, *6*, 13–13.
32. Azad, R.; Asadi-Aghbolaghi, M.; Kasaei, S.; Escalera, S. Dynamic 3D hand gesture recognition by learning weighted depth motion maps. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 1729–1740.
33. Bulbul, M.F.; Islam, S.; Ali, H. Human action recognition using MHI and SHI based GLAC features and collaborative representation classifier. *J. Intell. Fuzzy Syst.* **2019**, *36*, 3385–3401.
34. Weiyao, X.; Muqing, W.; Min, Z.; Yifeng, L.; Bo, L.; Ting, X. Human action recognition using multilevel depth motion maps. *IEEE Access* **2019**, *7*, 41811–41822.
35. Shekar, B.; Rathnakara Shetty, P.; Sharmila Kumari, M.; Mestetsky, L. Action recognition using undecimated dual tree complex wavelet transform from depth motion maps/depth sequences. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2019**, doi:10.5194/isprs-archives-XLII-2-W12-203-2019.
36. Bulbul, M.F.; Islam, S.; Ali, H. 3D human action analysis and recognition through GLAC descriptor on 2D motion and static posture images. *Multimed. Tools Appl.* **2019**, *78*, 21085–21111.
37. Al-Faris, M.; Chiverton, J.P.; Yang, Y.; Ndzi, D. Multi-view region-adaptive multi-temporal DMM and RGB action recognition. *Pattern Anal. Appl.* **2020**, *23*, 1587–1602.

38. Chen, C.; Jafari, R.; Kehtarnavaz, N. UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 168–172.
39. Youssef, C. Spatiotemporal representation of 3d skeleton joints-based action recognition using modified spherical harmonics. *Pattern Recognit. Lett.* **2016**, *83*, 32–41.
40. Hou, Y.; Li, Z.; Wang, P.; Li, W. Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *28*, 807–811.
41. Liu, Z.; Zhang, C.; Tian, Y. 3D-based deep convolutional neural network for action recognition with depth sequences. *Image Vis. Comput.* **2016**, *55*, 93–100.
42. Wang, P.; Wang, S.; Gao, Z.; Hou, Y.; Li, W. Structured images for RGB-D action recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 1005–1014.
43. Al-Obaidi, S.; Abhayaratne, C. Privacy Protected Recognition of Activities of Daily Living in Video. In Proceedings of the 3rd IET International Conference on Technologies for Active and Assisted Living (TechAAL 2019), London, UK, 25 March 2019; pp. 1–6.
44. Wang, L.; Ding, Z.; Tao, Z.; Liu, Y.; Fu, Y. Generative multi-view human action recognition. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6212–6221.
45. Liu, M.; Meng, F.; Chen, C.; Wu, S. Joint dynamic pose image and space time reversal for human action recognition from videos. Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8762–8769.
46. Lopez-Nava, I.H.; Muñoz-Meléndez, A. Human action recognition based on low-and high-level data from wearable inertial sensors. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719894532.
47. Haider, F.; Salim, F.A.; Postma, D.B.; Delden, R.v.; Reidsma, D.; van Beijnum, B.J.; Luz, S. A super-bagging method for volleyball action recognition using wearable sensors. *Multimodal Technol. Interact.* **2020**, *4*, 33.
48. Lemieux, N.; Noumeir, R. A hierarchical learning approach for human action recognition. *Sensors* **2020**, *20*, 4946.
49. Bai, Y.; Tao, Z.; Wang, L.; Li, S.; Yin, Y.; Fu, Y. Collaborative Attention Mechanism for Multi-View Action Recognition. *arXiv* **2020**, arXiv:2009.06599.
50. Ullah, A.; Muhammad, K.; Hussain, T.; Baik, S.W. Conflux LSTMs network: A novel approach for multi-view action recognition. *Neurocomputing* **2020**, *414*, 90–100.
51. Liu, Y.; Wang, L.; Bai, Y.; Qin, C.; Ding, Z.; Fu, Y. Generative View-Correlation Adaptation for Semi-supervised Multi-view Learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 318–334.
52. Ullah, A.; Muhammad, K.; Ding, W.; Palade, V.; Haq, I.U.; Baik, S.W. Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. *Appl. Soft Comput.* **2021**, *103*, 107102.
53. Kobayashi, T.; Otsu, N. Motion recognition using local auto-correlation of space-time gradients. *Pattern Recognit. Lett.* **2012**, *33*, 1188–1195.
54. Liu, K.; Ma, B.; Du, Q.; Chen, G. Fast motion detection from airborne videos using graphics processing unit. *J. Appl. Remote Sens.* **2012**, *6*, 061505.
55. Wright, J.; Ma, Y.; Mairal, J.; Sapiro, G.; Huang, T.S.; Yan, S. Sparse representation for computer vision and pattern recognition. *Proc. IEEE* **2010**, *98*, 1031–1044.
56. Tikhonov, A.N.; Arsenin, V.Y. Solutions of ill-posed problems. *N. Y.* **1977**, *1*, 30.
57. Golub, G.H.; Hansen, P.C.; O’Leary, D.P. Tikhonov regularization and total least squares. *SIAM J. Matrix Anal. Appl.* **1999**, *21*, 185–194.
58. Li, W.; Zhang, Z.; Liu, Z. Action recognition based on a bag of 3d points. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 9–14.
59. Lin, Y.C.; Hu, M.C.; Cheng, W.H.; Hsieh, Y.H.; Chen, H.M. Human action recognition and retrieval using sole depth information. In Proceedings of the 20th ACM international conference on Multimedia, Nara, Japan, 29 October–2 November 2012; pp. 1053–1056.
60. Gorelick, L.; Blank, M.; Shechtman, E.; Irani, M.; Basri, R. Actions as space-time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 2247–2253.
61. Blank, M.; Gorelick, L.; Shechtman, E.; Irani, M.; Basri, R. Actions as space-time shapes. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1, Beijing, China, 17–21 October 2005; Volume 2, pp. 1395–1402.
62. Vieira, A.W.; Nascimento, E.R.; Oliveira, G.L.; Liu, Z.; Campos, M.F. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In *Iberoamerican Congress on Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 252–259.
63. Bulbul, M.F.; Ali, H. Gradient local auto-correlation features for depth human action recognition. *SN Appl. Sci.* **2021**, *3*, 1–13.