3D Action Recognition Using Multi-scale Energy-based Global Ternary Image

Mengyuan Liu, *Student Member, IEEE,* Hong Liu†, *Member, IEEE,* Chen Chen, *Member, IEEE*

***Abstract*— This paper presents an effective multi-scale energy-based Global Ternary Image (GTI) representation for action recognition from depth sequences. The unique property of our representation is that it takes the spatial-temporal discrimination and action speed variations into account, intending to solve the problems of distinguishing similar actions and identifying the actions with different speeds in one goal. The entire method is carried out in two stages. In the first stage, consecutive depth frames are used to generate GTI features, which implicitly capture both interframe motion regions and motion directions. Specifically, each pixel in GTI represents one of three possible states, namely positive, negative and neutral, which indicate increased, decreased and same depth values, respectively. To cope with speed variations in actions, energy-based sampling method is utilized, leading to multi-scale energy-based GTI (E-GTI) features, where the multi-scale scheme can efficiently capture the temporal relationships among frames. In the second stage, all the E-GTI features are transformed by Radon Transform (RT) as robust descriptors, which are aggregated by the Bag-of-Visual-Words (BoVW) model as compact representation. Extensive experiments on benchmark datasets show that our representation outperforms state-of-the-art approaches, since it captures discriminating spatial-temporal information of actions. Due to the merits of energy-based sampling and RT methods, our representation shows robustness to speed variations, depth noise and partial occlusions.**

***Index Terms*—Action recognition, depth sequence, human-computer interaction**

I. Introduction

How to accurately recognize actions, e.g., hug, hand wave and smoke, in a cost-effective manner is one main challenge that confronts human-computer interaction, content-based video analysis and intelligent surveillance. Most previous works have used color cameras to record actions as RGB sequences and developed distinctive action representations for action analysis [1], [2], [3], [4]. However, action recognition using RGB sequences continues to be challenging because of problems such as severe changes in lighting conditions and cluttered backgrounds. Moreover, information loss in depth channel introduces ambiguity that renders difficulty in distinguishing similar actions.

With rapid advances of imaging technology in capturing depth information in real time, many works [6], [7], [8] solve action recognition problems by using depth data from depth cameras, particularly the cost-effective Microsoft Kinect RGB-D camera. Compared with conventional RGB data, depth data is more robust to changes in lighting conditions, because the depth value is estimated by infrared radiation without relating it to visible light [9]. Subtracting foreground from cluttered background is easier using depth data, as the confusing texture and color information from cluttered backgrounds are ignored [10]. In addition, RGB-D cameras provide depth maps with appropriate resolution and accuracy, which provide three-dimensional information on the structure of subjects/objects in the scene [11].

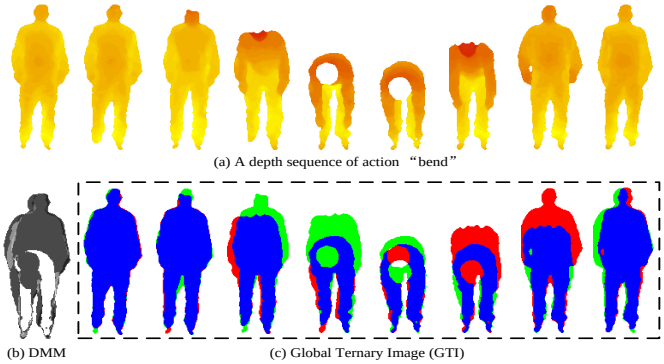


Figure 1: **Comparison between DMM and GTI.** (a) is a depth sequence of action "bend". (b) shows the DMM of the front view projection generated using all depth frames [5]. (c) shows the GTIs of the front view projection generated using consecutive depth frames. Pixels in red, green and blue colors denote positive, negative and neutral states, respectively. (Best viewed in color)

One of the most challenging tasks in 3D action recognition, e.g. action recognition using Kinect sensor, is to describe 3D motions from depth sequences which contain redundant data and noise. To this end, various representations of depth sequences have been developed, including Moving Pose (MP) [12], Histogram of 4D normals (HON4D) [13], Random Occupancy Pattern (ROP) [14] and depth motion map (DMM) [5]. Among these methods, DMM-based representations transform the action recognition problem from 3D to 2D and have been successfully applied to depth-based action recognition.

Motivation and Contributions: Our method is directly inspired by DMM-HOG [5], shown in Fig. 2 (a). In the work [5], depth maps were projected onto three orthogonal planes and a depth motion map (DMM) in each plane was generated by accumulating foreground regions through an entire sequence. Then, histogram of gradients (HOG) feature was used to describe each DMM. Finally, HOG features were concatenated as final representation. The success of DMM indicates that describing interframe motions is an efficient way to encode 3D action. However, DMM of an entire depth

M. Liu is with Faculty of Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China, e-mail: liumengyuan@pku.edu.cn.

H. Liu†, the corresponding author of this paper, is with full professor of Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China, e-mail: hongliu@pku.edu.cn.

C. Chen is with the Center for Research in Computer Vision at University of Central Florida, Orlando, FL 32816, USA e-mail: chenchen870713@gmail.com

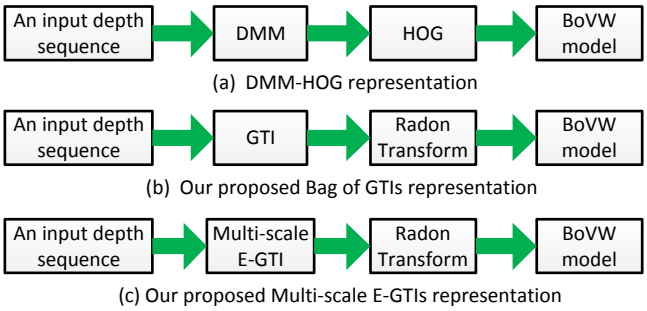


Figure 2: DMM-HOG [5] representation and our representations.

sequence can barely capture detailed temporal motion in a subset of depth frames. As a result, an old motion history may get overwritten when a more recent action occurs at the same point. An example is shown in Fig. 1 (b), where the limitation of DMM in capturing detailed motions is illustrated.

To this end, we propose a Global Ternary Image (GTI) feature, which outperforms DMM in capturing detailed frame-to-frame motion information including both motion regions and directions. As shown in Fig. 1 (c), it is observed that more detailed motion information (positive motion colored in red and negative motion colored in green) of the human body can be captured by GTIs than DMM. In previous work [5], HOG is used to describe the textures of DMM. Since GTI mainly contains shape information and lacks in texture information, we used Radon Transform (RT) instead to describe the shape of GTI. The pipeline of building Bag of GTIs representation is detailed in Fig. 2 (b). We observe that the speed variations of different performers affect the shape of GTI directly. Therefore, we propose an energy-based sampling method, which converts a depth sequence into multi-scale speed insensitive depth sequences. Based on these sequences, we develop a multi-scale energy-based GTIs representation, shown in Fig. 2 (c). Two main contributions are as follows:

- Multi-scale energy-based Global Ternary Image representation efficiently captures spatial-temporal discrimination of depth sequences, and it outperforms most state-of-the-art methods on benchmark datasets designed for 3D action and gesture recognition.
- The proposed representation shows robustness to common problems in real applications, i.e., speed variations, depth noise and partial occlusions, therefore it achieves best performances on modified MSRAction3D datasets, which involve above problems.

The remainder of this paper is as follows. Section II reviews related work. Section III presents Global Ternary Image and Section IV provides multi-scale Energy-based Global Ternary Image. Experimental results and comparisons are reported in Section V. Finally, Section VI concludes the paper.

II. Related Work

Action recognition methods using sole depth sensor are reviewed in this section. It is noted that fusing depth and other sensors like inertial sensor [15], [16], [17] can provide more abundant cues for analysis. According to the type of input data, 3D action recognition methods are roughly categorized into skeleton-based approaches and depth-based approaches. Taking adopted feature type into consideration, depth-based approaches are further categorized into local feature-based approaches and global feature-based approaches. Comprehensive reviews on 3D action recognition appear in [18], [19], [20].

Recent approaches built end-to-end systems by directly using original image/video as inputs to train neural networks [21], [22]. Specifically, differential Recurrent Neural Networks (RNN) [21] and part-aware Long Short-Term Memory (LSTM) [22] have been proposed to model temporal relationships among frames. Although high accuracies have been achieved, deep learning-based methods need large labeled data and time cost for training. It is noted that this work focuses on related hand-crafted feature-based approaches.

A. Skeleton-based approaches

Since actions can be denoted by movements of skeleton joints, related methods [23], [12], [24], [25], [26] represented motions by encoding 3D skeleton joint positions, estimated by tracking framework [27]. Yang *et al.* adopted the differences of joints in temporal and spatial domains to encode the dynamics of joints and then obtained EigenJoints by applying Principal Component Analysis (PCA) to joint differences [23]. The EigenJoints contain less redundancy and noise, when compared with original joints. Zanfir *et al.* provided a non-parametric Moving Pose (MP) framework, which considers more features like position, speed and acceleration of joints [12]. To ensure precision of estimated joints, Wang *et al.* incorporated temporal constraints and additional segmentation cues of consecutive skeleton joints for selecting K-best joint estimations [24]. Another way to improve the performance of skeleton joints is to associate local features with joints. This idea, termed as Actionlet Ensemble Model by Wang *et al.*, combines local occupancy pattern with 3D joints [25]. Pairwise relative positions of skeleton joints were also utilized in the work [25], because they are more discriminating and intuitive than previous skeleton joints-based features. Additionally, Luo *et al.* reduced the irrelevant information of pairwise skeleton joints feature and proposed a 3D joint feature, which selects one joint as reference and uses its differences to the remaining joints as features [26]. However, applications of skeleton-based approaches are limited, since skeleton data may be inaccurate when a person is indirectly standing toward the camera. Moreover, skeleton can be barely obtained in applications like hand gesture recognition.

B. Depth-based approaches

1) Local feature: Intuitively, surface normals reflect the shape of 3D objects. When human actions are treated as space-time pattern templates [28], the task of human action recognition is converted to 3D object recognition, therefore surface normals can be used to represent human actions [29], [13], [30]. Tang *et al.* formed a Histogram of Oriented Normal Vectors (HONV) as a concatenation of histograms of zenith and azimuthal angles to capture local distribution of the orientation of an object surface [29]. Oreifej *et al.* extended HONV to 4D space of time, depth and spatial coordinates, and provided a Histogram of Oriented 4D Normals (HON4D) to encode the surface normal orientation of human actions

[13]. HON4D jointly captures motion cues and dynamic shapes, therefore it shows more discriminating than previous approaches which separately encode the motion or shape information. To increase the robustness of HON4D against noise, Yang *et al.* grouped local hypersurface normals into polynormal, and aggregated low-level polynormals into Super Normal Vectors (SNV) [30].

Another type of local feature is called cloud points, which denotes human actions as clouds of local points. Li *et al.* extracted points from the contours of planar projections of 3D depth map, and employed an action graph to model the distribution of sampled 3D points [31]. Vieira *et al.* divided 3D points into same size of 4D grids, and applied Spatio-Temporal Occupancy Patterns (STOP) to encode these grids [32]. Wang *et al.* explored an extremely large sampling space of Random Occupancy Pattern (ROP) features and used a sparse coding method to encode these features [14].

Generally speaking, surface normals and cloud points show robustness against partial occlusions. When parts of features are destroyed by partial occlusions, the rests of local features are still useful to represent human actions. However, these local feature-based methods ignore the global constrains among points therefore they are not distinctive to classify human actions with similar local structures.

2) Global feature: As a traditional method of extracting motions, frame difference method calculates the differences between consecutive frames to generate motion regions. By accumulating these motion regions across a whole sequence, Boblick *et al.* proposed a Motion Energy Image (MEI) to represent where motion has occurred in an RGB sequence [33]. A Motion History Image (MHI) was also proposed in [33], where the intensity of each pixel in MHI is a function of temporal history information at that point.

By incorporating an additional dimension of depth, Azary *et al.* extended MHI to define a Motion Depth Surface (MDS), which captures most recent motions in the depth direction as well as within each frame [34]. To make full use of depth information, Yang *et al.* projected depth maps onto three orthogonal planes and generated a depth motion map (DMM) in each plane by accumulating foreground regions through an entire sequence [5]. Based on the concept of DMM, Chen *et al.* proposed an improved version, which stacks the depth values across an entire depth sequence for three orthogonal planes [35], [36]. Yang *et al.* used DMM to train deep convolutional neural networks (CNNs) [37]. On the basis of DMM, a DMM-Pyramid architecture was proposed to partially keep the temporal ordinal information lost in DMM. Zhang *et al.* proposed Edge Enhanced Depth Motion Map (E^2DMM) to balance the information weighing between shape and motion [38]. Additionally, they employed a dynamic temporal pyramid to segment the depth video sequence to address temporal structure information of dynamic hand gestures.

Generally speaking, DMM-based representations, which are able to effectively transform the action recognition problem from 3D to 2D, have achieved promising accuracies on the task of depth-based action recognition. However, interframe motions are directly accumulated in previous works. In other words, detailed motions (i.e. motion shapes and motion directions) have not been extensively explored. To solve this problem, we propose a new representation, which preserves distinctive interframe motion cues and encodes global temporal constrains among depth frames. Moreover, our representation shows robustness to speed variations, depth noise and partial occlusions.

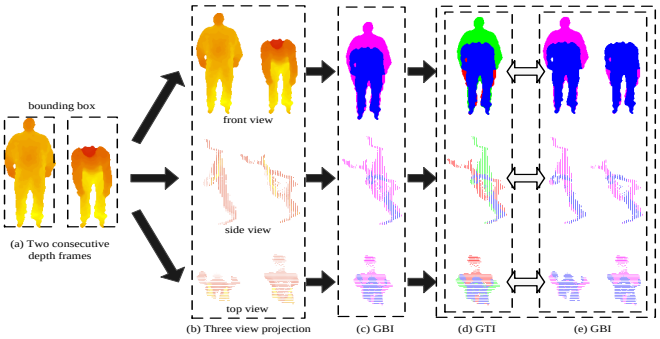


Figure 3: **Extraction of GBI and GTI.** (c) shows GBIs which denote motion regions (colored in pink). (d) shows GTIs which denote both motion regions and motion directions (pixels in green stand for negative motion and pixels in red stand for positive motion). (e) shows that each GTI can be described by two GBIs. (Best viewed in color)

III. Global Ternary Image

In this section, we firstly introduce the concept of Global Binary Image (GBI), which is designed to represent motion regions between two consecutive depth frames. Then, we extend GBI to Global Ternary Image (GTI), which is designed to represent both motion regions and motion directions. Further, GBI and GTI are transformed by Radon Transform (RT) to form compact and robust feature vectors.

Suppose $\mathcal{I}$ denote a depth sequence containing a 3D action, which is formulated as:

$$\mathcal{I} = \{I^1, ..., I^i, ..., I^N\}, \quad s.t.\ i \in (1, ..., N), \tag{1}$$

where N is the total number of depth frames and I^i is the i-th frame. The pipeline of extracting GBIs from two consecutive depth frames, i.e., I^i and I^{i-1}, is shown in Fig. 3 (a)-(c).

The depth value of each pixel in a frame shows two useful properties. First, the 3D shape information of human body can be inferred by the spatial distribution of depth values. Second, the changes in depth value across frames provide motion information in the depth direction. To make full use of depth information, each depth frame is projected onto three orthogonal planes:

$$I^i \rightarrow \{\, map_v^i \,\}, \quad s.t.\ v \in \{\, f, s, t \,\}, \tag{2}$$

where map_v^i is the projected map of the i-th depth frame on the v view; f, s, t stand for the front, side and top views.

On each projected map, background (i.e. zero) region is discarded and the bounding box of foreground (i.e. non-zero) region is selected as the region of interest. As shown in Fig. 3 (a), the bounding box is the smallest rectangle, which contains all regions that an actor can ever reach. To achieve scale invariance, foregrounds in bounding boxes are normalized to their respective sizes, which are fixed according to previous work [36]. The normalization eliminates the effect of different heights and motion extents of different performers also. After

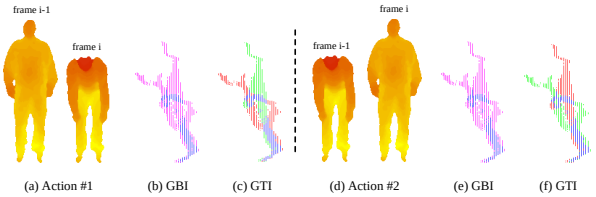


Figure 4: **Comparison between GBI and GTI.** Two depth frames are used to denote action #1 (a) and action #2 (d). GBI (b) and GTI (c) of the side view projection are generated for action #1. GBI (e) and GTI (f) of the side view projection are generated for action #2. (Best viewed in color)

these steps, the i-th depth map generates three 2D maps, one each on the front, side, and top views, i.e. map_f^i, map_s^i, map_t^i.

For each view, we detect depth value changes and obtain binary maps to indicate motion regions. In the following discussion, each binary map here is called a GBI. The value of GBI_v^i on location j is defined as:

$$GBI_v^i(j) = \begin{cases} 1, & if\ map_v^{i-1}(j) > 0 \wedge map_v^i(j) = 0 \\ 1, & if\ map_v^{i-1}(j) = 0 \wedge map_v^i(j) > 0 \\ 0, & otherwise, \end{cases} \quad (3)$$

where $map_v^i(j)$ is the depth value of map_v^i on location j. Here, we refer a pixel with non-zero depth value as a foreground point and refer a pixel with zero depth value as a background point. Under this assumption, Formula $map_v^{i-1}(j) > 0 \wedge map_v^i(j) = 0$ indicates that the pixel on location j jumps from the background to the foreground and Formula $map_v^{i-1}(j) = 0 \wedge map_v^i(j) > 0$ indicates that the pixel on location j jumps from the foreground to the background. In other words, the state of a pixel changing between background and foreground indicates the occurrence of motion. A neutral state is detected on a pixel when the depth value (non-zero) of a pixel grows bigger or changes to a smaller value (non-zero). This phenomenon indicates that GBI suffers less from depth noise, which severely changes original depth values.

Besides motion regions captured by GBI, motion directions also play an important role in describing motions. Two similar actions, i.e. action #1 in Fig. 4 (a) and action #2 in Fig. 4 (d), are taken as an example. Obviously, their corresponding GBIs, i.e. GBI in Fig. 4 (b) and GBI in Fig. 4 (e), are nearly the same, indicating similar motion regions (pink pixels in Fig. 4 (b) and Fig. 4 (e)). It is observed that the main difference between two actions are the motion directions. Therefore, GTI is proposed to encode both motion regions and motion directions, by adding directional information to GBI. GTIs in Fig. 4 (c) and (f) are different for two actions, indicating dissimilar motion directions. In other words, GTI can capture both motion regions and motion directions, therefore showing superior motion description power than GBI.

Similar to the step of formulating GBI_v^i, the value of corresponding GTI_v^i on location j is defined as:

$$GTI_v^i(j) = \begin{cases} +1, & if\ map_v^{i-1}(j) > 0 \wedge map_v^i(j) = 0 \\ -1, & if\ map_v^{i-1}(j) = 0 \wedge map_v^i(j) > 0 \\ 0, & otherwise, \end{cases} \quad (4)$$

where Formula $map_v^{i-1}(j) > 0 \wedge map_v^i(j) = 0$ indicates positive motion (red pixels in Fig. 3 (d)) and Formula $map_v^{i-1}(j) = 0 \wedge map_v^i(j) < 0$ indicates negative motion (green pixels in Fig. 3 (d)). It is noted that each GTI can be described as two GBIs, which are shown in Fig. 3 (e).

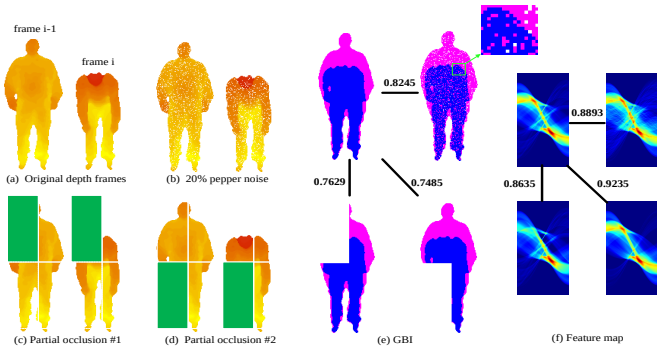


Figure 5: **The effect of Radon Transform.** Radon Transform describes GBIs of the front view projection generated using (a) original depth frames, (b) depth frames added by 20% pepper noise, (c) depth frames partially occluded by occlusion #1 (simulated by ignoring pixels located in the green region), (d) depth frames partially occluded by occlusion #2. Number in (e) and (f) denotes correlation coefficient. (Best viewed in color)

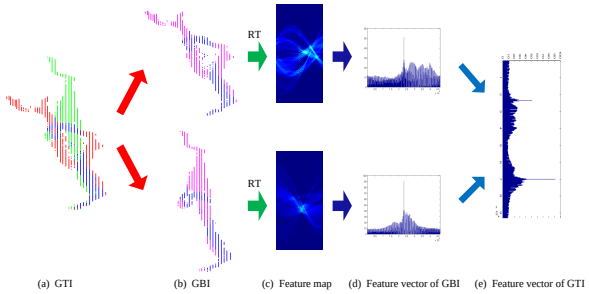


Figure 6: **Feature extraction from GTI.** (b) shows two GBIs, denoting positive and negative motion regions of the GTI in (a). (c) shows feature maps generated by describing GBIs using Radon Transform. Feature maps in (c) are converted to feature vectors in (d), which are concatenated to describe the GTI in (a). (Best viewed in color)

With the directional information from GTI, we can easily distinguish similar actions in Fig. 4, where Fig. 4 (c) shows an action of "a person bends the head to the waist" and Fig. 4 (f) shows an action of "a person unbends the body". These observations indicate that GTI is able to efficiently capture the details (e.g., regions and directions) of 3D motions.

GBI and GTI have shown robustness to depth value changes (from one non-zero value to another) caused by depth noise. However, they still suffer from space-time discontinuities in depth data and partial occlusions. Similar to previous work [39], pepper noises are added to original depth data to simulate the space-time discontinuity. In Fig. 5 (e), pepper noise brings fake motion regions and holes to GBI. Similar to previous work [14], partial occlusions are simulated by ignoring a portion of depth data. In Fig. 5 (e), the shapes of GBIs are dramatically changed by occlusions. To solve these problems, Radon Transform is introduced to convert GBIs to robust feature maps. Fig. 5 (e) and (f) show that the correlation coefficients between pairwise feature maps are larger than that of the corresponding pairwise GBIs. In other words, feature maps of GBIs share much similar appearances than GBIs, therefore showing stronger robustness to pepper noise and partial occlusions.

Mathematically speaking, Radon Transform in two dimensions is the integral transform consisting of the integral of a function over straight lines [40]. In other words, Radon Transform can find the projection of a shape on any given line.

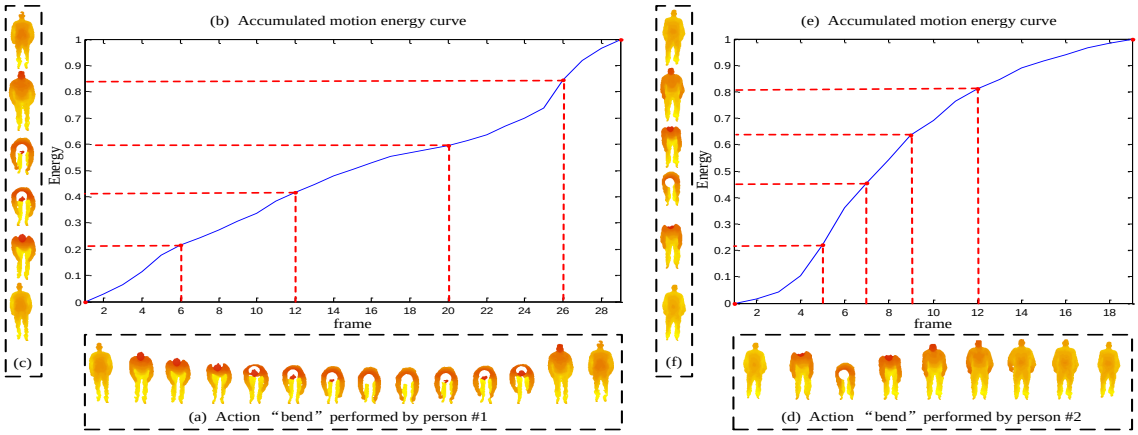


Figure 7: **Illustration of energy-based sampling method.** Action "bend" performed by person #1 and person #2 are shown in (a) and (d), respectively. (b) and (e) are the accumulated motion energy curves, calculated by accumulating frame-to-frame motion energy. Depth frames in (c) and (f) are sampled from (a) and (d), respectively. The sampling criterion is to keep frame-to-frame motion energy of the sampled sequence nearly the same. (Best viewed in color)

Given a compactly supported continuous function $f(x,y)$ on $\mathbb{R}^2$, the Radon Transform is defined as:

$$\mathscr{R}\{f(x,y),\theta\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\cdot \delta(xcos\theta + ysin\theta - \rho)dxdyd\rho, \tag{5}$$

where δ is the Dirac delta function, $\rho \in [-\infty, +\infty]$, and $\theta \in [0,\pi]$. When $f(x,y)$ stands for an image of W in width and H in height, ρ is limited to $\left[\lfloor -\frac{\sqrt{W^2+H^2}}{2}\rfloor, \lceil \frac{\sqrt{W^2+H^2}}{2}\rceil\right]$.

Let GBI_v^i denote a GBI, the corresponding feature vector generated by applying Radon Transform is formulated as:

$$R_v^i = \left[\mathscr{R}\{GBI_v^i, \theta_p\}\right]_{p=1}^{P}, \tag{6}$$

where $\theta_p \in [0\ \pi)$; each θ_p establishes a line on which the GBI is projected; P stands for the total number of projections; θ_p equals to $\frac{p}{P}\cdot\pi$. Fig. 6 shows the pipeline of extracting feature vector from a GTI GTI_v^i, which can be described as two GBIs: $+GTI_v^i\cdot(GTI_v^i > 0)$ and $-GTI_v^i\cdot(GTI_v^i < 0)$. Then, Formula 6 is applied to describe each GBI as a feature map, which is further reshaped as a feature vector. Let W and H denote the width and height of the GBI. Then, the dimension of the feature vector is fixed as $\lceil\sqrt{W^2+H^2}\rceil\cdot P$. Finally, the feature vector of GTI is calculated by concatenating feature vectors of corresponding two GBIs.

IV. Energy-based Global Ternary Image and Multi-scale Scheme

Different performers have different habits, which increase the intra-varieties among same type of actions. As shown in Fig. 7 (a) and (d), action "bend" is performed by person #1 and person #2, generating two depth sequences with different speeds. Since GBI and GTI are designed to describe interframe motions, speed variations will directly affect appearances of GBI and GTI. In this section, it is observed that the state of human pose in a sequence is related to motion energy. When a person changes his or her pose, from one to another, the motion energy extracted from two poses is a stable value, which is unrelated to speed. Therefore, we select frames from the original sequence to form a new sequence, in which the motion energies between consecutive frames are nearly the same. In this way, the sampled sequence suffers less from the effect of speed variations. Correspondingly, an energy-based sampling method is proposed to sample frames from original depth sequences. As shown in Fig. 7 (c) and (f), the sampled sequences are similar to each other, indicating slight effect from speed variations. The GTI extracted from sampled sequences is termed as Energy-based GTI (E-GTI), which inherits the merits of GTI and shows robustness to speed variations. Following paragraphs mainly focus on the energy-based sampling method.

Algorithm 1: Energy-based sampling method

Input: depth sequence $\mathcal{I} = \{I^i\}_{i=1}^N$, number of frames N, M
Output: sampled sequence S_M

$S^1 \leftarrow I^1, S^M \leftarrow I^N$;
$E^1 \leftarrow 0$;
for $i = 2; i \le N$ **do**
  $e \leftarrow 0$;
  for $v \in \{f, s, t\}$ **do**
    for $\forall j \in GBI_v^i$ **do**
      $GBI_v^i(j) \leftarrow$ Formula 3;
      $e \leftarrow e + num\{GBI_v^i(j)\}$;
  $E^i \leftarrow E^{i-1} + e$;
$m \leftarrow 2, i \leftarrow 2$;
while $m \le M-1$ **do**
  while $i \le N$ **do**
    if $\left(\frac{E^N}{M-1}\cdot(m-1)\right) \le E^i$ **then**
      $S^m \leftarrow I^i$;
      $m \leftarrow m+1$;
      break;
    $i \leftarrow i+1$;
return $S_M = \{S^m\}_{m=1}^M$;

Given a depth sequence with N frames, the accumulated motion energy on the i-th frame is defined as:

$$E^i = \sum_{j=2}^{i}\sum_{v\in\{f,s,t\}} num\{GBI_v^j\}, \tag{7}$$

where $num\{\cdot\}$ returns the number of non-zero elements in a binary map. For simplicity of expression, E^1 is set to zero.

In Algorithm 1, frames from a given depth sequence are

selected to construct a sampled sequence with M frames. The pipeline of such a construction can be divided into two steps. First, the first and the last frames of the given sequence are selected as the starting and ending frames of the sampled sequence. Second, $M-2$ frames are selected to make sure that motion energies between consecutive frames are nearly equal. As shown in Fig. 7 (b) and (e), accumulated motion energies are calculated for both actions. Following Algorithm 1, we obtain sampled sequences with parameter M setting to six for example. It can be seen that the intra-varieties between sampled sequences, i.e., Fig. 7 (c) and (f), are much smaller than those between original sequences, i.e., Fig. 7 (a) and (d).

To encode temporal information, Laptev *et al.* proposed a pyramid-based representation to take into account the rough temporal order of a sequence [1]. Yang *et al.* observed that it is inflexible to handle action speed variations by evenly subdividing a video along the time axis [30]. Therefore, they defined a concept of motion energy to adaptively divide a sequence into several temporal segments with equal motion energy. Specifically, the starting and ending frames of each segment are adaptively selected by using motion energy. Then, the segment is constructed by all frames from original sequence, which locate between the starting and ending frames. This pipeline indicates that the problem of speed variations is still unsolved in each segment. Different from previous work [30], we sample frames from original sequence using motion energy and construct sampled sequences directly using sampled frames. In this way, each frame of the sampled sequence is related to the motion energy, therefore our sampled sequences show robustness to speed variations.

In the field of image retrieval, Bag-of-Visual-Words (BoVW) model is widely used to obtain a compact representation from local features. Here, we represent a depth sequence $\mathcal{I}$ by a set of GBIs $R=\{R^i\}_{i=2}^N$, where R^i is defined as:

$$R^i = [\, R^i_f,\ \ R^i_s,\ \ R^i_t\,], \tag{8}$$

which concatenates feature vectors from three projection views. During the training stage of BoVW, local features are randomly selected from training set and then clustered into K "words" using clustering method, such as K-means [41]. During the testing stage, BoVW model finds the corresponding "word" for each feature in the feature set R and then uses a histogram of "words" as a simple representation of $\mathcal{I}$:

$$\begin{aligned} B^{\mathcal{I}}_{GBI} &= \mathcal{B}\{R, K\} \\ &= \mathcal{B}\{\{R^i\}_{i=2}^N, K\}, \end{aligned} \tag{9}$$

where function $\mathcal{B}$ stands for performing all steps of BoVW model. To remove the effect of the number of local features, the above representation is further normalized as:

$$B^{\mathcal{I}}_{GBI} = \frac{\mathcal{B}\{\{R^i\}_{i=2}^N, K\}}{||\mathcal{B}\{\{R^i\}_{i=2}^N, K\}||_2}, \tag{10}$$

where $||\cdot||_2$ calculates the l_2 norm. Following similar steps, Radon Transform and BoVW model are applied to generate a representation $B^{\mathcal{I}}_{GTI}$, when the depth sequence $\mathcal{I}$ is represented by a set of low-level GTIs.

Suppose the depth sequence $\mathcal{I}$ is sampled to form a sampled sequence S_M with M frames, which can be represented by a set of low-level E-GTIs. Similarly, a representation $B^{\mathcal{I}}_{E-GTI}$ is formed to describe S_M. However, a sampled sequence S_M only preserves one certain scale of motion from the original sequence $\mathcal{I}$. As shown in Fig. 8, three sampled sequences, i.e., S_3, S_6 and S_8, are sampled from an action "bend". It is noted that the parameter M is set to 3, 6 and 8 for example. As can be seen, three sampled sequences record different scales of motions. Specifically, motions in larger scale are captured by sequence #1 and sequence #2, meanwhile motions in smaller scale are captured by sequence #3. To capture different scales of motion information, we sample multi-scale depth sequences to give a detailed description of $\mathcal{I}$. We set parameter M in Algorithm 1 to $M_1, ..., M_L$, which produces a number of L sampled sequences, i.e., $S_{M_1}, ..., S_{M_L}$, from sequence $\mathcal{I}$. Let $B^{S_{M_L}}_{E-GTI}$ denote the representation of S_{M_L}. By concatenating representations of all sampled sequences, we obtain representation-level fused representation as:

$$B^{\mathcal{I}}_{E-GTI} = \left[B^{S_{M_1}}_{E-GTI},\ \cdots,\ B^{S_{M_L}}_{E-GTI}\right], \tag{11}$$

which captures multi-scale motions of sequence $\mathcal{I}$. Moreover, the multi-scale scheme implicitly captures temporal relationships among frames.

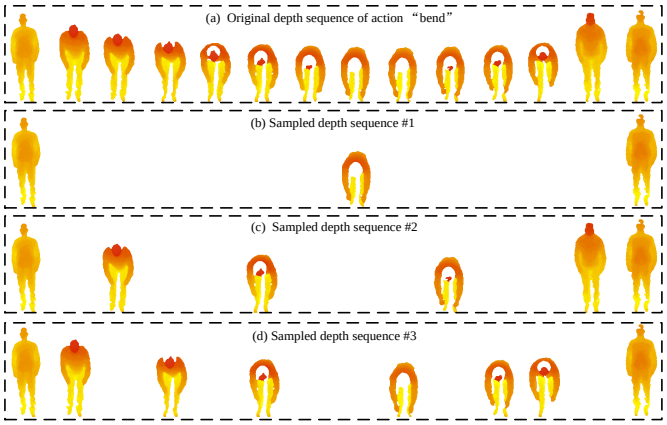


Figure 8: **Illustration of multi-scale sampled sequences.** (a) is an original sequence. (b), (c) and (d) are sampled depth sequences, which record different scales of motions. (Best viewed in color)

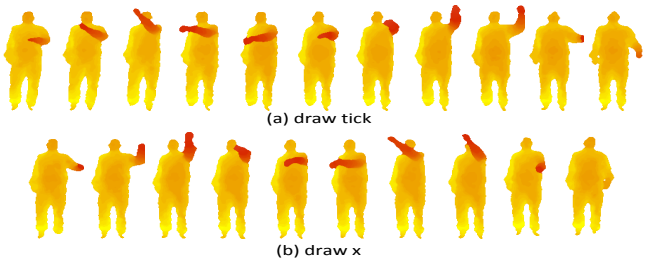


Figure 9: Action snaps from MSRAction3D dataset.

V. Experiments and Discussions

A. Experiments with MSRAction3D dataset

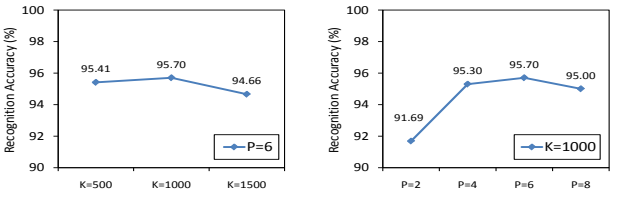
1) ***Dataset****:* MSRAction3D dataset [31] stands out as one of the most widely used depth datasets in literature [43]. It contains 20 actions: "high arm wave", "horizontal arm wave", "hammer", "hand catch", "forward punch", "high throw",

Table I: Selection of K, P on the training samples of MSRAction3D dataset with 10-fold procedure. Training samples are defined in previous work [42].

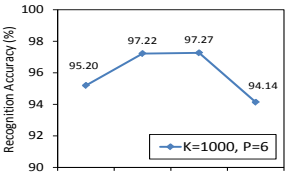
Accuracy (%)	P = 2	P = 4	P = 6	P = 8
K = 500	95.26	97.28	97.61	97.57
K = 1000	96.57	98.66	**99.00**	98.61
K = 1500	97.95	98.64	98.97	98.97

Table II: Selection of L on the training samples of MSRAction3D dataset with 10-fold procedure. Training samples are defined in previous work [42].

L	1	2	3	4
Accuracy (%)	98.59	99.23	**99.85**	98.64



(a) Effect of K on Bag of GTIs model (b) Effect of P on Bag of GTIs model



(b) Effect of L on Multi-scale E-GTIs model

Figure 10: Evaluation of K, P, L on MSRAction3D dataset with protocol of work [42].

"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 & throw". Each action is performed two or three times by ten subjects facing the depth camera, resulting in 567 depth sequences. As shown in Fig. 9, actions like "draw x" and "draw tick" are similar except for slight differences between movements of one hand.

*2) **Settings**:* The recognition is conducted using a non-linear SVM with a homogeneous Chi2 kernel [44] and parameter "gamma", which decides the degree of homogeneity of the kernel, is set to 0.8. We use the "sdca" solver for SVM, besides other default parameters are set according to the vlfeat library [1]. Similar to previous work [36], the bounding boxes of front, side and top views are resized to fixed sizes of 102×54, 102×75 and 75×54.

*3) **Parameter selection**:* Let K be the cluster number for K-means and P be the number of projections for Radon Transform. We use a baseline representation $B^{\mathcal{I}}_{GTI}$ to select proper K and P. This representation is generated by performing Bag of GTIs model on original depth sequences. In Table I, parameter K changes from 500 to 1500 at an interval of 500, and parameter P changes from 2 to 8 at an interval of 2. We select these parameters on the training samples of MSRAction3D dataset with 10-fold procedure. It is noted that the samples performed by subjects $\#1, 3, 5, 7, 9$ are defined as training samples [42]. Highest accuracy of 99.00% is obtained when K and P are set to 1000 and 6, respectively. In following, we set default values of parameters K and P as 1000 and 6, which also work well on other datasets.

We select proper L for multi-scale representation $B^{\mathcal{I}}_{E-GTI}$.

[1]Simple code to use non-linear SVM for classification can be found in http://www.vlfeat.org/applications/caltech-101-code.html

Table III: **Comparison between our method and related works on the MSRAction3D dataset with protocol of work [42].** "L" is short for local depth feature. "G" is short for global depth feature. "S" is short for skeleton feature. "D" is short for depth learning method.

Methods	Accuracy (%)	Year	Type
Motion Depth Surface [34]	78.48	2013	G
STOP [32]	84.80	2012	L
ROP+SC [14]	86.20	2012	L
STK-D+Local HOPC [45]	86.50	2016	L
LSTM [21]	87.78	2015	D+S
Actionlet Ensemble [42]	88.20	2014	L+S
HON4D [13]	88.89	2013	L
H3DF [46]	89.45	2015	L
LSGF [47]	90.76	2016	L
HOG3D+LLC [48]	90.90	2015	L
Moving Pose [12]	91.70	2013	S
dRNN [21]	92.03	2015	D+S
Hierarchical 3D Kernel [49]	92.73	2015	L
4DCov+Sparse Collab. [50]	93.01	2014	G
MMMP [51]	93.10	2015	L+S
Multi-fused features [52]	93.30	2016	G+S
Super Normal Vector [30]	93.45	2016	L
Depth Context [53]	94.28	2015	L
Hierarchical RNN [54]	94.49	2015	D+S
MBS [55]	95.20	2015	L
Range-Sample [56]	95.62	2014	L
Ker-RP-RBF [57]	96.90	2015	S
Key-Pose-Motifs [58]	97.44	2016	S
3D-CNN+DMM-Cube [37]	86.08	2014	D+G
DMM-HOG [5]	88.73	2012	G
2D-CNN+DMM-Pyramid [37]	91.21	2014	D+G
WHDMM [59]	92.73	2015	G
DMM-LBP-DF [36]	93.00	2015	G
2D-CNN+WHDMM [59]	**100.00**	2015	D+G
Bag of GTIs	95.70	2016	G
Multi-scale E-GTIs	97.27	2016	G

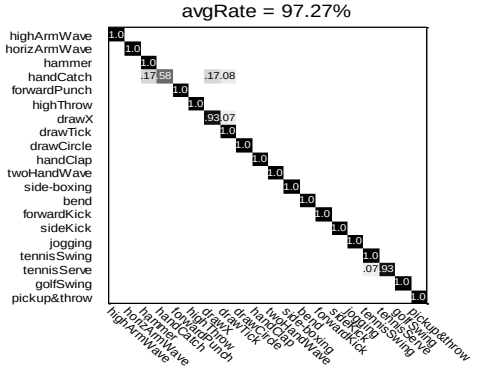


Figure 11: Confusion matrix on MSRAction3D with protocol of work [42].

In MSRAction3D dataset, the average frame number of depth sequences is around 40 frames, therefore we set the maximum value of S_{M_L} to 40. It is noted that when a sequence contains less than 40 frames, we interpolate the original sequence to 40 frames as a preprocessing step. To simplify the evaluation, we convert original sequences into sampled sequences with 10, 20, 30, 40 frames. Multi-scale representation $B^{\mathcal{I}}_{E-GTI}$ is denoted as the representation-level fusion of sampled sequences. Take scale L=3 as an example, we test possible combinations of three sequences from the four sampled sequences, and report the best accuracy as the performance of this scale. As shown in Table II, we achieve the highest accuracy of 99.85% when L is set to 3. Therefore, we set default value of parameter L as 3 for MSRAction3D dataset. In following, we select proper L for different datasets in similar way.

With the protocol of work [42], we test the effect of parameters K, P, L on our models. Fig. 10 shows that selected

Table IV: **Comparison between our method and related works on the MSRAction3D dataset with protocol of work [31].** The original work is colored in blue.

Methods	Accuracy (%)	Year	Type
Bag of 3D Points [31]	74.70	2010	L
Motion Depth Surface [34]	78.48	2013	G
Key Poses+Pyramid [60]	95.14	2016	S
MBS [55]	97.00	2015	L
MMMP [51]	98.20	2015	L+S
DMM-HOG [5]	91.63	2012	G
WHDMM [59]	95.62	2015	G
TPDM-SPHOG [61]	96.14	2015	G
2D-CNN+WHDMM [59]	**100.00**	2015	D+G
Bag of GTIs	97.47	2016	G
Multi-scale E-GTIs	99.16	2016	G

parameters achieve optimal performances in certain scopes.

*4) **Comparison with related works**:* We adopt two types of cross-validation methods on this dataset. First is the overall cross subject accuracy regardless of subsets [42], and second is the average cross subject performance on three action subsets defined in previous work [31]. Table III and IV show the results. According to Section II, related works can be divided into different types, denoted as "L", "G", "S" and "D". "L" is short for local depth feature; "G" is short for global depth feature; "S" is short for skeleton feature; "D" is short for deep learning method; "+" denotes the combination of two methods, e.g. "L+S" combines local depth and skeleton features.

In Table III, the proposed Bag of GTIs and Multi-scale E-GTIs methods with default parameters achieve accuracies of 95.70% and 97.27%. As shown in Fig. 11, most types of actions are correctly classified. Our method outperforms traditional global depth features, e.g., 4DCov+Sparse Collab. [50] and DMM-LBP-DF [36]. This result shows that our method encodes more abundant motion and temporal information. Our method outperforms all local depth features, e.g., Range-Sample [56] and Depth Context [53]. The reason is that global feature can encode the global relationships among body parts, which are ignored by local features. Some skeleton features, e.g., Ker-RP-RBF [57] and Key-Pose-Motifs [58], achieve high accuracies. However, skeleton joints can be accurately estimated only when action performers directly face the camera. These joints are usually not reliable when action performers are not in upright position (e.g. sit on a seat or lie on a bed). What is worse, partial occlusions seriously affect the accuracy of skeleton extraction method and thus limit the wide usage of skeleton-based methods. Therefore, our method, which directly operates on depth values, is more suitable for practical applications.

Our method is directly comparable with DMM-HOG [5], which uses histogram of gradients (HOG) feature to encode DMM. The proposed Bag of GTIs outperforms DMM by 6.97%, which verifies that detailed interframe motion information captured by GTI is more distinctive than accumulated motion information captured by DMM feature. The proposed Multi-scale E-GTIs outperforms Bag of GTIs by 1.57%, which shows that the temporal information captured by the multi-scale scheme benefits the recognition of similar actions.

We also compare our method with deep learning methods, which can be roughly divided into three categories, i.e., RNN/LSTM, 3D-CNN and 2D-CNN. Based on skeleton features, RNN/LSTM-based methods, i.e., LSTM [21], dRNN [21] and Hierarchical RNN [54], have achieved high accuracies. However, these methods usually over overemphasize the temporal evaluations and ignore the discriminating power of spatial information [62]. Combining 3D-CNN and DMM, 3D-CNN+DMM-Cube [37] only achieves 86.08%. The reason is that it requires a large scale of labeled depth sequence samples to train a good 3D CNN model. Combining 2D-CNN and DMM, 2D-CNN+WHDMM [59] has achieved high accuracy. To evaluate the effect of extracting deep features from hand-crafted features, we use the BoVW model to aggregate WHDMM features as depth sequence representation, which achieves accuracy of 92.73%. The 2D-CNN+WHDMM outperforms WHDMM by 7.27%, indicating that 2D-CNN can improve the discriminating power of hand-crafted features. Without relying on CNN model, our method achieves competitive accuracies with 2D-CNN+WHDMM method. Moreover, our method shows robustness to depth noise, partial occlusions and speed variations, which have not been explored in previous works, e.g. 2D-CNN+WHDMM.

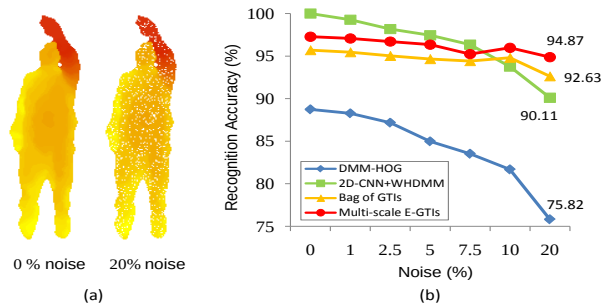


Figure 12: Evaluation of robustness to depth noise. (a) Depth frames affected by 0% and 20% percentage of pepper noise. (b) Recognition results on MSRAction3D dataset with different percentages of pepper noise.

B. Experiments with modified MSRAction3D datasets

*1) **Depth noise**:* As illustrated in Fig. 12 (a), we follow work [39] to simulate depth discontinuities in depth sequences by adding pepper noise in varying percentages (of the total number of image pixels) to depth images. In Fig. 12 (b), both Bag of GTIs and Multi-scale E-GTIs achieve more than 92% accuracies with different percentages of pepper noise. Compared with 2D-CNN+WHDMM [59], our method achieves better results when the percentage of pepper noise is larger than 10%. These improvements indicate that Radon Transform reduces the intra-variations brought by depth discontinuities, which is illustrated in Fig. 5 (e) and (f).

*2) **Partial occlusions**:* We follow work [14] to simulate partial occlusions using sequences from MSRAction3D dataset. Each volume of the depth sequence is divided into two parts along x, y and t dimensions, resulting in eight subvolumes. The occlusion is simulated by ignoring the depth data in one of the subvolumes. Totally, eight new datasets are generated, and some of their snaps are shown in Fig. 13.

To verify the effect of Radon Transform (RT), we use a pixel value-based descriptor to describe GTI instead. Suppose GTI_v^i denote a GTI, we firstly convert it to a pair of GBIs: $+GTI_v^i \cdot (GTI_v^i > 0)$ and $-GTI_v^i \cdot (GTI_v^i < 0)$. Then, pixel

Table V: Evaluation of the robustness to partial occlusions.

Dataset / Accuracy (%)	ROP [14]	ROP+SC [14]	DMM-HOG [5]	2D-CNN+WHDMM [59]	**Bag of GTIs** (without RT)	**Bag of GTIs**	**Multi-scale E-GTIs**
MSRAction3D	85.92	86.20	88.73	**100.00**	88.20	95.70	97.27
Occlusion #1	83.05	86.17	73.63	90.84	73.16	90.55	**92.31**
Occlusion #2	84.18	86.50	52.38	92.31	51.33	93.45	**95.24**
Occlusion #3	78.76	80.09	79.85	89.38	80.16	91.30	**93.41**
Occlusion #4	82.12	85.49	78.75	87.91	79.87	88.61	**91.58**
Occlusion #5	84.48	87.51	64.47	86.08	62.08	87.39	**90.11**
Occlusion #6	82.46	87.51	71.06	89.38	71.66	89.07	**93.04**
Occlusion #7	80.10	83.80	68.50	91.58	70.89	91.52	**93.77**
Occlusion #8	85.83	86.83	76.56	94.87	80.56	94.13	**97.07**

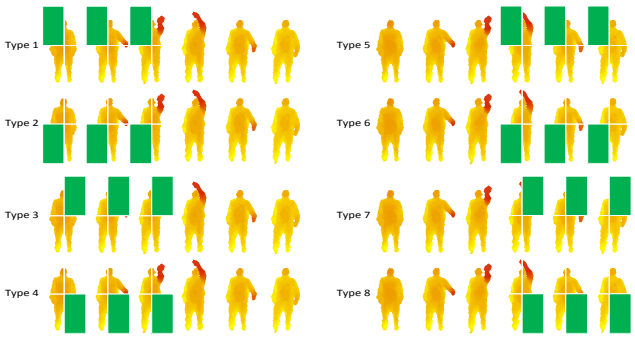


Figure 13: **Eight types of occluded depth sequences.** Each sequence is denoted as six frames for example.

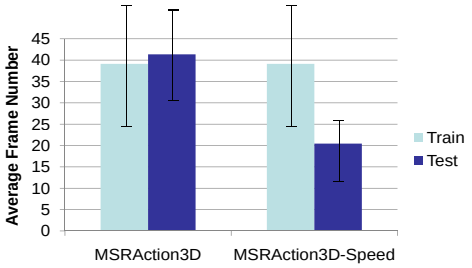


Figure 14: Speed differences between training and testing sets.

values of GBIs are directly concatenated as the descriptor of GTI, which is named as Bag of GTIs (without RT).

Table V compares the robustness of different methods to partial occlusions. Bag of GTIs outperforms Bag of GTIs (without RT) on all cases, which verifies that RT can properly suppress the intra-varieties, e.g., partial occlusions, as illustrated in Fig. 5 (e) and (f). A comparison of the performances between our method and Random Occupancy Pattern (ROP) feature [14] shows that our method achieves higher accuracy with all kinds of occlusions than ROP feature. It is noted that sparse coding (SC) can improve the robustness of a given feature to occlusions [14]. Without using sparse coding, our method still outperforms "ROP+SC", in face of most types of occlusions. Compared with DMM-HOG [5] and 2D-CNN+WHDMM [59], our method achieves better results with all types of occlusions. These improvements indicate that RT can efficiently improve the robustness of a given feature to partial occlusions.

*3) **Speed variations**:* Speed variations bring intra-varieties among same types of actions. In Fig. 14, we use average frame number of action sequence as an indicator of action speed. Obviously, the speed differences between training and testing sets of MSRAction3D dataset are quite small. While, training on a large set of actions performed in various speeds can reduce the effect of speeds. To eliminate the effect of training data, we evaluate the robustness of our method against

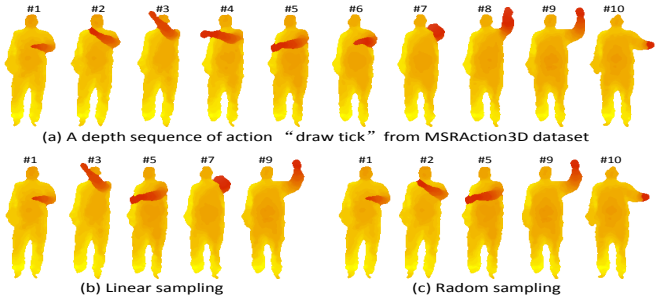


Figure 15: Comparison between linear sampling and random sampling.

Table VI: Evaluation of the robustness of speed variations.

Accuracy (%)	MSRAction3D	MSRAction3D-Speed
DMM-HOG [5]	88.73	76.19
2D-CNN+WHDMM [59]	**100.00**	88.64
Bag of GTIs	95.70	87.17
Multi-scale E-GTIs	97.27	**91.30**

various speeds on an MSRAction3D-Speed [2] dataset, which contains totally different speeds between training and testing sets. Specifically, we reserve all the sequences performed by subjects #1, 3, 5, 7, 9 and randomly select half the number of frames for sequences performed by subjects #2, 4, 6, 8, 10. Based on the original time order, the selected frames are catenated to form new sequences. Fig. 14 shows that the difference in average frames between the training and testing sets of the new dataset has been enlarged. Since random sampling method is used, many key frames may be ignored in new sequences, which makes action recognition more challenging. Comparing linear sampling method with our random sampling method (see Fig. 15), we infer that action speeds in MSRAction3D-Speed dataset may change dramatically in a non-linear manner.

Table VI compares the robustness of different methods to speed variations. Multi-scale E-GTIs achieves 97.27% on MSRAction3D dataset and achieves 91.30% on MSRAction3D-Speed dataset. These results indicate that MSRAction3D-Speed dataset is more challenge than MSRAction3D dataset. Compared with DMM-HOG [5] and 2D-CNN+WHDMM [59], our method achieves better results on the MSRAction3D-Speed dataset. The reason is that our method alleviates the effect of speed variations by energy-based sampling method, which is illustrated in Fig. 7.

C. Experiments with DHA dataset

DHA dataset [63] contains action types extended from Weizmann dataset [67] which is widely used to evaluate action recognition methods using RGB sequences. It contains 23 action categories: "arm-curl", "arm-swing", "bend", "front-box", "front-clap", "golf-swing", "jack", "jump", "kick", "leg-

[2]Our collected MSRAction3D-Speed dataset can be found in https://github.com/CvDatasets/depthDatasets.git

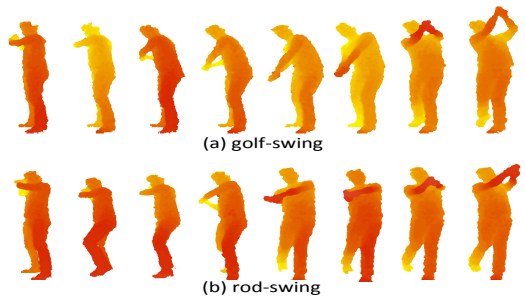


Figure 16: Action snaps from DHA dataset.

Table VII: **Comparison between our method and related works on the DHA dataset with protocol of work [63].** Original work is colored in blue.

Methods	Accuracy (%)	Year	Type
DMHI-Gist/CRC [64]	86.00	2015	G
D-STV/ASM [63]	86.80	2012	L
DMHI-AHB-Gist/CRC [64]	90.50	2015	G
D-DMHI-PHOG [65]	92.40	2015	G
DMPP-PHOG [65]	95.00	2015	G
DMM-HOG [5]	86.50	2012	G
2D-CNN+WHDMM [59]	92.86	2015	D+G
Bag of GTIs	91.92	2016	G
Multi-scale E-GTIs	**95.44**	2016	G

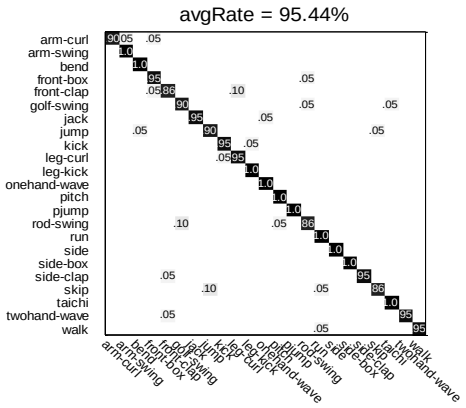


Figure 17: Confusion matrix on DHA dataset with protocol of work [63].

curl", "leg-kick", "one-hand-wave", "pitch", "p-jump", "rod-swing", "run", "skip", "side", "side-box", "side-clap", "tai-chi", "two-hand-wave", "walk". Each action is performed by 21 people (12 males and 9 females), resulting in 483 depth sequences. We use an extended version of DHA dataset where six additional action categories are involved. In Fig. 16, "golf-swing" and "rod-swing" share similar motions that involve moving hands from one side up to the other side. A few more such similar pairs can be found, like "leg-curl" and "leg-kick", "run" and "walk", etc. Bounding boxes of front, side and top views are resized to fixed sizes of 102×54, 102×75, 75×54. Other parameters are the same with MSRAction3D dataset.

In Table VII, Lin *et al.* [63] achieves 86.80% on the original DHA dataset. By encoding interframe constraints among space-time volumes in a multi-scale way, our method achieves higher accuracy even on the extended DHA dataset. DMM-based methods, e.g. DMM-HOG [5] and 2D-CNN+WHDMM [59], suffer from the effect of speed variations, therefore these methods do not work well on DHA dataset, which contains more severe speed variations than MSRAction3D dataset. Since our method shows robustness to speed variations, we achieve best performance on this dataset. Confusion matrix

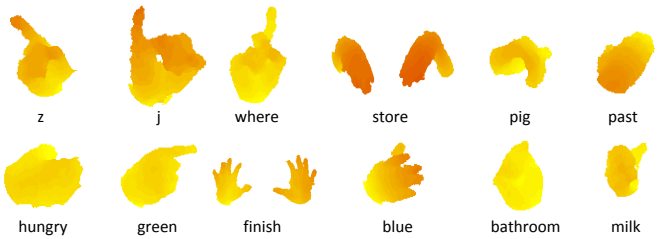


Figure 18: Action snaps from MSRGesture3D dataset.

Table VIII: **Comparison between our method and related works on the MSRGesture3D dataset with protocol of work [14].**

Methods	Accuracy (%)	Year	Type
Motion Depth Surface [34]	85.42	2013	G
Random Occupancy Pattern [14]	88.50	2012	L
HON4D [13]	92.45	2013	L
4DCov+Sparse Collab. [50]	92.89	2014	G
HOG3D+LLC [48]	94.10	2015	L
MBS [55]	94.70	2015	L
Super Normal Vector [30]	94.74	2016	L
H3DF [46]	95.00	2015	L
Depth Gradients+RDF [66]	95.29	2014	L+S
Hierarchical 3D Kernel [49]	95.66	2015	L
STK-D+Local HOPC [45]	96.23	2016	L
DMM-HOG [5]	88.20	2012	G
E^2DMM [38]	90.50	2013	G
3D-CNN+DMM-Cube [37]	92.25	2014	D+G
2D-CNN+DMM-Pyramid [37]	94.35	2014	D+G
DMM-LBP-DF [36]	94.60	2015	G
Bag of GTIs	96.42	2016	G
Multi-scale E-GTIs	**98.80**	2016	G

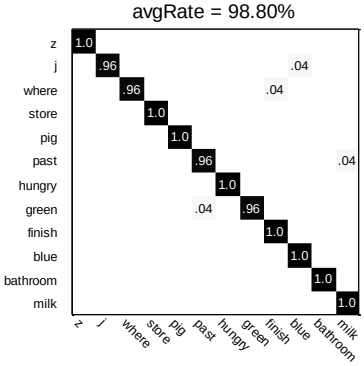


Figure 19: Confusion matrix on MSRGesture3D dataset with protocol of work [14].

of our method on the DHA dataset, with highest accuracy of 95.44%, is shown in Fig. 17, where similar actions like 'golf-swing" and "rod-swing" contain small ambiguities.

D. Experiments with MSRGesture3D dataset

MSRGesture3D dataset [14] is a hand gesture dataset. It contains 12 gestures, defined by American Sign Language: "z", "j", "where", "store", "pig", "past", "hungry", "green", "finish", "blue", "bathroom" and "milk". Each gesture is performed two or three times by each subject, resulting in 333 depth sequences. In Fig. 18, actions like "past" and "hungry" are similar, because both actions contain similar poses of palm. Moreover, self-occlusions bring extra challenges. The bounding boxes of front, side and top views are resized to fixed sizes of 118×133, 118×29 and 29×133. Other parameters are the same with MSRAction3D dataset.

In Table VIII, DMM-HOG [5] achieves accuracy of 88.20% on this dataset. Recent works [37], [36] further enhance

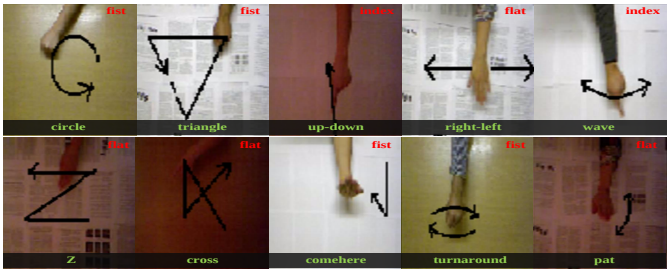


Figure 20: Original action snaps from SKIG dataset.

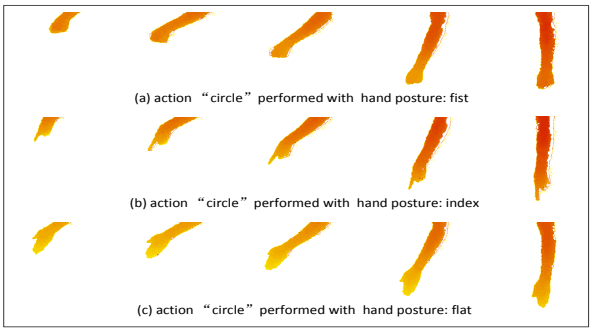


Figure 21: Action snaps from SKIG dataset, where backgrounds are removed.

Table IX: **Comparison between our method and related works on the SKIG dataset with protocol of work [68].** Original work is colored in blue.

Methods	Accuracy (%)	Year	Type
HOG/HOF [1]	72.10	2008	L
HOG3D [70]	75.40	2008	L
LFF+SPP [71]	81.10	2015	L
RGGP+RGBD [68]	88.70	2013	G
LFF+SPP+RGBD [71]	93.70	2016	L
4DCov+Sparse Collab. [50]	93.80	2014	G
Bag of GTIs	90.87	2016	G
Multi-scale E-GTIs	**93.88**	2016	G

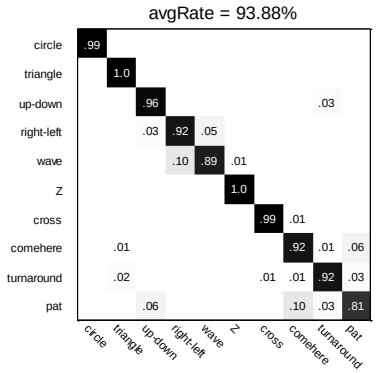


Figure 22: Confusion matrix on SKIG dataset with protocol of work [68].

the original DMM by using deep features or LBP features. DMM-LBP-DF [36] achieves an accuracy of 94.60%, which outperforms all previous DMM-based methods. Our Bag of GTIs outperforms [36] by 1.82%, which verifies that GTI has superior descriptive power than DMM in capturing motion regions and motion directions. Using representation-level fusion, we achieve state-of-the-art result of 98.80% (see Fig. 19), which is even 2.57% higher than the most recent local depth feature-based method, i.e., STK-D+Local HOPC [45]. This improvement shows that our global feature can properly capture the temporal and motion information of 3D gestures.

E. Experiments with SKIG dataset

SKIG dataset [68] contains 1080 hand-gesture depth sequences. It contains ten gestures, which comprise "circle", "triangle", "up-down", "right-left", "wave", "Z", "cross", "comehere", "turnaround" and "pat". All gestures are performed with hand (i.e., fist, flat and index) by six subjects under two different illumination conditions (i.e., strong and poor light) and against three backgrounds (i.e., white plain paper, wooden board and paper with characters). This dataset is utilized to test the robustness of our method against pose and illumination variations. To eliminate the effect of background in the original SKIG dataset, we apply the foreground extraction method [69] to extract hand regions. Several snaps are shown in Fig. 21, where the cluttered backgrounds are removed. The bounding boxes of front, side and top views are resized to fixed sizes of 118×133, 118×29 and 29×133. Other parameters are the same with MSRAction3D dataset.

In Table IX, Bag of GTIs achieves an accuracy of 90.87% on this dataset. This result shows that the intra-varieties, caused by different hand poses, can be properly tackled with Radon Transform. Therefore, our method outperforms original work [68], which may suffer from the effect of pose variations. In work [50], a collaborative sparse classifier is presented, taking advantage of 4DCov descriptor laying on a specific manifold topology. Using representation-level fusion, we achieve state-of-the-art result of 93.88%. As shown in Fig. 22, "right-left" and "wave" have large ambiguities. This reason is that both actions have similar movement of "moving one hand from one side to another". For the similar reason, there also exists ambiguities among "comehere" and "pat".

F. Evaluation of Global Ternary Image

The representation of GBI, named $B^{\mathcal{I}}_{GBI}$, obtains an accuracy of 90.33% on MSRAction3D dataset (shown in Fig. 23). $B^{\mathcal{I}}_{GBI}$ achieves 1.6% higher accuracy than DMM-HOG [5], since more abundant motion information is preserved in GBIs. To further verify the significance of extracting interframe motions, we use shape information to form representation which is denoted by $B^{\mathcal{I}}_{shape}$. Specifically, each depth frame is projected onto three views. For each view, we extract the foreground region which stands for the shape. Following the same steps that are used in forming $B^{\mathcal{I}}_{GBI}$, we can obtain $B^{\mathcal{I}}_{shape}$ using Radon Transform and BoVW model. Only 61.50% recognition accuracy is achieved by $B^{\mathcal{I}}_{shape}$ on MSRAction3D dataset, which is 28.83% lower than $B^{\mathcal{I}}_{GBI}$. This indicates that inter-frame motions can efficiently reduce the ambiguity contained in shape information.

The representation of GTI, named $B^{\mathcal{I}}_{GTI}$ obtains an accuracy of 95.70% on MSRAction3D dataset (shown in Fig. 23). Compared with GBIs, GTIs additionally contain directional information about motions; therefore, $B^{\mathcal{I}}_{GTI}$ achieves 5.37% higher accuracy than $B^{\mathcal{I}}_{GBI}$. To further verify the significance of encoding directions, we compare $B^{\mathcal{I}}_{GTI}$ and $B^{\mathcal{I}}_{GBI}$ on a new dataset, named MSRAction3D-Order. We double the sequences from MSRAction3D dataset by inverting their temporal order. In other words, the new dataset contains double the number of action types, where each type corresponds to an opposite type. As expected, $B^{\mathcal{I}}_{shape}$ and $B^{\mathcal{I}}_{GBI}$ perform

Table X: Evaluation of single scale and multi-scale structures.

Accuracy (%)	0	1	2	3	4	1+2	1+3	1+4	2+3	2+4	3+4	1+2+3	1+2+4	1+3+4	2+3+4	1+2+3+4
MSRAction3D	95.70	85.18	94.79	94.78	95.20	93.89	94.45	95.58	97.22	97.14	96.36	95.58	96.68	96.27	**97.27**	94.14
MSRAction3D-Speed	87.17	83.00	86.52	85.72	86.70	87.44	90.80	89.64	90.31	90.16	89.11	**91.30**	90.06	90.85	91.03	90.57
DHA	91.92	84.26	89.02	93.58	91.51	91.51	92.96	91.92	93.16	92.75	93.58	94.40	93.58	95.03	94.61	**95.44**
MSRGesture3D	96.42	93.71	96.42	96.42	97.32	97.32	98.21	97.91	98.21	97.32	98.21	98.51	98.51	**98.80**	98.21	**98.80**
SKIG	90.87	84.71	89.48	91.85	91.48	90.83	92.86	92.21	92.63	92.59	93.05	92.68	92.91	93.37	93.65	**93.88**

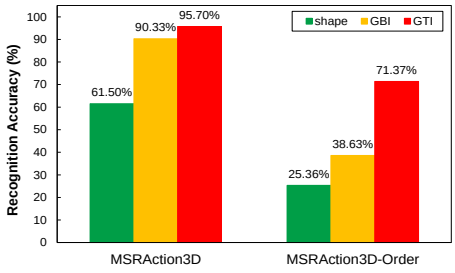


Figure 23: Evaluation of GBI and GTI.

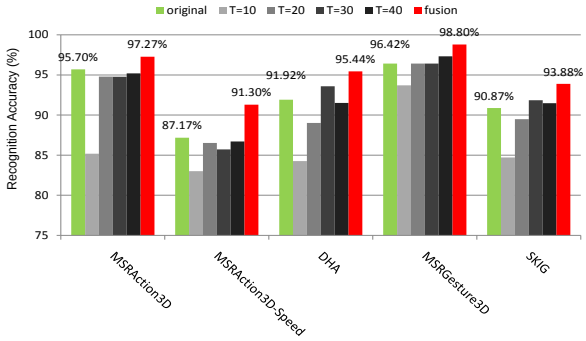


Figure 24: Evaluation of single scale and multi-scale structures.

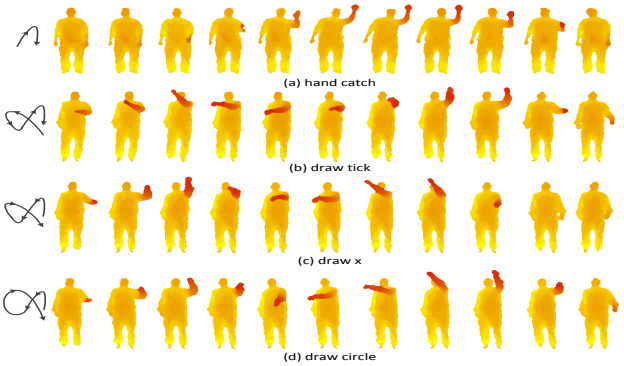


Figure 25: Difficult cases in MSRAction3D dataset.

worse on MSRAction3D-Order dataset than on MSRAction3D dataset. This is because one action type and its opposite type contain similar motion regions, which bring extra challenges to the task of classification. $B^{\mathcal{I}}_{GTI}$ achieves an accuracy of 71.37%, which is higher than that of both $B^{\mathcal{I}}_{shape}$ and $B^{\mathcal{I}}_{GBI}$. This improvement is justifiable because GTI captures directional information, which is essential to distinguish one action type from its opposite type.

Generally speaking, $B^{\mathcal{I}}_{GTI}$ and $B^{\mathcal{I}}_{GBI}$ outperform $B^{\mathcal{I}}_{shape}$ on both MSRAction3D and MSRAction3D-Order datasets, which emphasizes the distinctive power of interframe motions, rather than shape information. Meanwhile, $B^{\mathcal{I}}_{GTI}$ outperforms $B^{\mathcal{I}}_{GBI}$, especially on MSRAction3D-Order dataset, which reflects the effect of directional information in describing motions.

G. Evaluation of multi-scale structure

As single sampled sequence ignores much motion information of original sequence, we extract multiple sampled sequences and use multi-scale E-GTIs to describe the original sequence. In MSRAction3D dataset, the average frame number of depth sequences is around 40 frames, therefore we set the maximum value of S_{M_L} to 40. To simplify the evaluation, we convert original sequences into sampled sequences with 10, 20, 30, 40 frames. We use $B^{\mathcal{I}}_{GTI}$, $B^{S_{10}}_{E-GTI}$, $B^{S_{20}}_{E-GTI}$, $B^{S_{30}}_{E-GTI}$, $B^{S_{40}}_{E-GTI}$ (short for 0,1,2,3,4) (see Table X) to describe the original sequence and the four corresponding sampled sequences. Multi-scale representation is denoted as $B^{\mathcal{I}}_{E-GTI}$, which is the representation-level fusion of sampled sequences. Accuracy of 97.27% is achieved by $B^{\mathcal{I}}_{E-GTI}$ on MSRAction3D dataset, which is only 1.57% higher than that of $B^{\mathcal{I}}_{GTI}$. Meanwhile, from a comparison between $B^{\mathcal{I}}_{E-GTI}$ and $B^{\mathcal{I}}_{GTI}$, the accuracy is found to have improved by 4.13% on MSRAction3D-Speed dataset. These improvements lead to two conclusions. First, converting original sequences to sampled sequences is beneficial for capturing multiple temporal information. Second, the effect of this conversion can be enlarged especially on datasets whose sequences in training and testing sets have big gaps in the distribution of speeds. Using $B^{\mathcal{I}}_{E-GTI}$, we obtain higher accuracies than those of $B^{\mathcal{I}}_{GTI}$ on DHA, MSRGesture3D and SKIG datasets. The improvements are illustrated in Fig. 24, which shows the efficiency of multi-scale E-GTIs.

H. Difficult cases

In Fig. 25, four similar actions from MSRAction3D dataset are illustrated to show the advantages and disadvantages of our method. (1) Although the motion direction of action "draw x" is similar to that of action "draw circle", our method can detect the difference by encoding their individual interframe motion regions. (2) Although the motion regions of action "draw x" are similar to those of action "draw tick", our method can distinguish them by using directional information of motions. (3) The action "hand catch" can be regard as a sub-action of action "draw tick". In this case, misclassification may happen, because histograms of local features, generated by BoVW model, are similar for these actions. To solve this problem, deep learning methods, e.g., RNN/LSTM, could be used to further explore deep structures of our hand-crafted features.

I. Computation time

We test the computation time of our method with the default parameters of $K = 1000$ and $P = 6$. The average computational time required for extracting a GTI is 0.0363 seconds on a 2.5GHz machine with 8GB RAM, using Matlab R2012a. The calculation time for applying Radon Transform

on a GTI is 0.0019 seconds. The overall computational time for calculating a feature vector of GTI is about 0.0381 seconds. It is noted that each GTI can be extracted from consecutive frames and then transformed by Radon Transform, which shows that the feature extraction step for a depth sequence can be conducted in parallel on a CPU/GPU.

VI. Conclusions and Future Work

In this paper, we present a multi-scale energy-based Global Ternary Image representation, which efficiently encodes both spatial-temporal information of 3D actions. Compared with Depth Motion Map (DMM)-based approaches, our method preserves detailed interframe motion regions and directions. Moreover, the temporal relationships among frames are captured by a multi-scale scheme. Therefore, our method outperforms DMM and achieves comparable results with state-of-the-art methods on benchmark datasets designed for 3D action and gesture recognition tasks. With the developments of energy-based sampling and Radon Transform methods, our method shows robustness against speed variations, depth noise and partial occlusions, which are common yet unsolved problems for real applications. The robustness of our method is evaluated on a series of modified MSRAction3D datasets, where we achieve best performances.

Recent deep learning-based methods usually extract deep features from raw depth sequences. While, problems like speed variations, depth noise and partial occlusions bring ambiguities to deep features. Our method directly tackles with these problems, and the generated features can be further explored by deep neural networks to increase their discriminative power. Previous works like Moving Pose [12] and Hierarchical RNN [54] rely on skeleton joints, which can be accurately estimated only when action performers directly face the camera, thus limiting the wide usage of skeleton-based methods. Therefore, our method, which directly operates on depth values, is more suitable for real applications. Future work focuses on developing real-time action recognition system for monitoring the behavior, e.g., fall down and wave hands, of the elders. To deal with real world scenarios for action recognition, it also calls for creating depth action datasets that incorporate realistic problems, e.g. noise and occlusions.

VII. Acknowledgements

This work is supported by National High Level Talent Special Support Program, National Natural Science Foundation of China (NSFC, No.61340046,61673030,U1613209), Specialized Research Fund for the Doctoral Program of Higher Education (No.20130001110011), Natural Science Foundation of Guangdong Province (No.2015A030311034), Scientific Research Project of Guangdong Province (No.2015B010919004).

References

[1] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, pp. 1–8, 2008.

[2] H. Liu, M. Liu, and Q. Sun, "Learning directional co-occurrence for human action classification," in *ICASSP*, pp. 1235–1239, 2014.

[3] X. Ma, F. Bashir, A. A. Khokhar, and D. Schonfeld, "Event analysis based on multiple interactive motion trajectories," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 397–406, 2009.

[4] M. Liu, H. Liu, Q. Sun, T. Zhang, and R. Ding, "Salient pairwise spatio-temporal interest points for real-time activity recognition," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 14–29, 2016.

[5] X. Yang, C. Zhang, and Y. L. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *ACM MM*, pp. 1057–1060, 2012.

[6] C. Chen, M. Liu, B. Zhang, J. Han, J. Jiang, and H. Liu, "3D action recognition using multi-temporal depth motion maps and fisher vector," in *IJCAI*, 2016.

[7] H. Liu, Q. He, and M. Liu, "Human action recognition using adaptive hierarchical depth motion maps and gabor filter," in *ICASSP*, 2017.

[8] C. Chen, B. Zhang, Z. Hou, J. Jiang, M. Liu, and Y. Yang, "Action recognition from depth sequences using weighted fusion of 2d and 3d auto-correlation of gradients features," *Multimed. Tools and Appl.*, pp. 1–19, 2016.

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

[10] M. Harville, G. Gordon, and J. Woodfill, "Foreground segmentation using adaptive mixture models in color and depth," in *IEEE Workshop on Detection and Recognition of Events in Video*, pp. 3–11, 2001.

[11] B. Ni, G. Wang, and P. Moulin, "Rgbd-hudaact: A color-depth video database for human daily activity recognition," in *Consumer Depth Cameras for Computer Vision*, pp. 193–208, 2013.

[12] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection," in *ICCV*, pp. 2752–2759, 2013.

[13] O. Oreifej and Z. Liu, "Hon4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *CVPR*, pp. 716–723, 2013.

[14] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *ECCV*, pp. 872–885, 2012.

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

[16] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimed. Tools and Appl.*, pp. 1–21, 2015.

[17] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 1, pp. 51–61, 2015.

[18] H. Cheng, L. Yang, and Z. Liu, "A survey on 3D hand gesture recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1659–1673, 2015.

[19] J. K. Aggarwal and X. Lu, "Human activity recognition from 3D data: A review," *Pattern Recogn. Lett.*, vol. 48, no. 1, pp. 70–80, 2014.

[20] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, "Rgb-d-based action recognition datasets: A survey," *Pattern Recogn.*, vol. 60, pp. 86–105, 2016.

[21] V. Veeriah, N. Zhuang, and G. J. Qi, "Differential recurrent neural networks for action recognition," in *ICCV*, pp. 4041–4049, 2015.

[22] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3d human activity analysis," in *CVPR*, 2016.

[23] X. Yang and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *CVPRW*, pp. 14–19, 2012.

[24] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *CVPR*, pp. 915–922, 2013.

[25] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *CVPR*, pp. 1290–1297, 2012.

[26] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *ICCV*, pp. 1809–1816, 2013.

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

[28] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, 2007.

[29] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *ACCV*, pp. 525–538, 2013.

[30] X. Yang and Y. L. Tian, "Super Normal Vector for human activity recognition with depth cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, 10.1109/TPAMI.2016.2565479, 2016.

[31] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *CVPRW*, pp. 9–14, 2010.

[32] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. Campos, "Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 252–259, 2012.
[33] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, 2001.
[34] S. Azary and A. Savakis, "Grassmannian sparse representations and motion depth surfaces for 3D action recognition," in *CVPRW*, pp. 492–499, 2013.
[35] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *J. Real-Time Image Pr.*, pp. 1–9, 2013.
[36] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in *WACV*, pp. 1092–1099, 2015.
[37] R. Yang and R. Yang, "Dmm-pyramid based deep architectures for action recognition with depth cameras," in *ACCV*, pp. 37–49, 2014.
[38] C. Zhang and Y. Tian, "Edge enhanced depth motion map for dynamic hand gesture recognition," in *CVPRW*, pp. 500–505, 2013.
[39] S. Jetley and F. Cuzzolin, "3D activity recognition using motion history and binary shape templates," in *ACCVW*, pp. 129–144, 2014.
[40] S. R. Deans, "Applications of the radon transform," *Wiley Interseience Publications*, 1983.
[41] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *ACM MM*, pp. 1469–1472, 2010.
[42] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 914–927, 2014.
[43] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "A discussion on the validation tests employed to compare human action recognition methods using the MSR Action3D dataset," *arXiv:1407.7390*, 2014.
[44] A. Zisserman and Oxford, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 480–492, 2012.
[45] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Histogram of oriented principal components for cross-view action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 10.1109/TPAMI.2016.2533389, 2016.
[46] C. Zhang and Y. Tian, "Histogram of 3D facets: A depth descriptor for human action and hand gesture recognition," *Comput. Vis. Image Und.*, vol. 139, pp. 29–39, 2015.
[47] E. Zhang, W. Chen, Z. Zhang, and Y. Zhang, "Local surface geometric feature for 3D human action recognition," *Neurocomputing*, vol. 208, pp. 281–289, 2016.
[48] H. Rahmani, Q. H. Du, A. Mahmood, and A. Mian, "Discriminative human action classification using locality-constrained linear coding," *Pattern Recogn. Lett.*, vol. 72, pp. 62–71, 2015.
[49] Y. Kong, B. Satarboroujeni, and Y. Fu, "Hierarchical 3D kernel descriptors for action recognition using depth sequences," in *FG*, pp. 1–6, 2015.
[50] P. Cirujeda and X. Binefa, "4DCov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences," in *International Conference on 3D Vision*, pp. 657–664, 2014.
[51] A. Shahroudy, T.-T. Ng, Q. Yang, and G. Wang, "Multimodal multipart learning for action recognition in depth videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, p. DOI 10.1109/TPAMI. 2015.2505295, 2015.
[52] A. Jalal, Y. H. Kim, Y. J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recogn.*, vol. 61, pp. 295–308, 2016.
[53] M. Liu and H. Liu, "Depth Context: A new descriptor for human activity recognition by using sole depth sequences," *Neurocomputing*, vol. 175, pp. 747–758, 2015.
[54] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, pp. 1110–1118, 2015.
[55] Y. Yang, B. Zhang, L. Yang, C. Chen, and W. Yang, "Action Recognition Using Completed Local Binary Patterns and Multiple-class Boosting Classifier," in *ACPR*, pp. 336–340, 2015.
[56] C. Lu, J. Jia, and C. K. Tang, "Range-Sample depth feature for action recognition," in *CVPR*, pp. 772–779, 2014.
[57] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li, "Beyond covariance: Feature representation with nonlinear kernel matrices," in *ICCV*, pp. 4570–4578, 2015.
[58] C. Wang, Y. Wang, and A. L. Yuille, "Mining 3D key-pose-motifs for action recognition," in *CVPR*, pp. 2639–2647, 2016.
[59] P. Wang, W. Li, Z. Gao, and J. Zhang, "Action Recognition From Depth Maps Using Deep Convolutional Neural Networks," *IEEE Trans. Human-Mach. Syst.*, vol. 46, pp. 498–509, 2015.
[60] E. Cippitelli, E. Gambi, S. Spinsante, and F. Florez-Revuelta, "Human Action Recognition Based on Temporal Pyramid of Key Poses Using RGB-D Sensors," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 510–521, 2016.
[61] H. Xu, E. Chen, C. Liang, L. Qi, and L. Guan, "Spatio-Temporal Pyramid Model based on depth maps for action recognition," in *International Workshop on Multimedia Signal Processing*, pp. 1–6, 2015.
[62] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *ACM MM*, pp. 102–106, 2016.
[63] Y.-C. Lin, M.-C. Hu, W.-H. Cheng, Y.-H. Hsieh, and H.-M. Chen, "Human action recognition and retrieval using sole depth information," in *ACM MM*, pp. 1053–1056, 2012.
[64] A. A. Liu, W. Z. Nie, Y. T. Su, L. Ma, T. Hao, and Z. X. Yang, "Coupled hidden conditional random fields for rgb-d human action recognition," *Signal Process.*, vol. 112, pp. 74–82, 2015.
[65] Z. Gao, H. Zhang, G. Xu, and Y. Xue, "Multi-perspective and multi-modality joint representation and recognition model for 3D action recognition," *Neurocomputing*, vol. 151, pp. 554–564, 2015.
[66] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," in *WACV*, pp. 626–633, 2014.
[67] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, vol. 2, pp. 1395–1402, 2005.
[68] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *IJCAI*, pp. 1493–1500, 2013.
[69] C. Zhang, X. Yang, and Y. Tian, "Histogram of 3D facets: A characteristic descriptor for hand gesture recognition," in *FG*, pp. 1–8, 2013.
[70] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *BMVC*, pp. 1–10, 2008.
[71] M. Yu, L. Liu, and L. Shao, "Structure-preserving binary representations for RGB-D action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1651–1664, 2016.

Mengyuan Liu received the B.E. degree in intelligence science and technology in 2012, and is working toward the Ph.D. degree in the School of EE&CS, Peking University (PKU), China. His research interests include action recognition using depth and skeleton data. He has published articles in Neurocomputing, MTA, ROBIO, ICIP, ICASSP, 3DV and IJCAI.

Hong Liu received the Ph.D. degree in mechanical electronics and automation in 1996, and serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. Prof. Liu has been selected as Chinese Innovation Leading Talent supported by National High-level Talents Special Support Plan since 2013. He is also the Director of Open Lab on Human Robot Interaction, PKU, his research fields include computer vision and robotics, image processing, and pattern recognition. Dr. Liu has published more than 150 papers and gained Chinese National Aero-space Award, Wu Wenjun Award on Artificial Intelligence, Excellence Teaching Award, and Candidates of Top Ten Outstanding Professors in PKU. He is an IEEE member, vice president of Chinese Association for Artificial Intelligent (CAAI), and vice chair of Intelligent Robotics Society of CAAI. He has served as keynote speakers, co-chairs, session chairs, or PC members of many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO, IEEE SMC and IIHMSP, recently also serves as reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing, and IEEE Trans. on PAMI.

Chen Chen received the B.E. degree in automation from Beijing Forestry University, Beijing, China, in 2009 and the M.S. degree in electrical engineering from Mississippi State University, Starkville, in 2012 and the Ph.D. degree in the Department of Electrical Engineering at the University of Texas at Dallas, Richardson, TX in 2016. He is a Post-Doc in the Center for Research in Computer Vision at University of Central Florida (UCF). His research interests include signal and image processing, pattern recognition and computer vision.