2016 Fourth International Conference on 3D Vision

Energy-based Global Ternary Image for Action Recognition Using Sole Depth Sequences

Mengyuan Liu
Key Laboratory of Machine Perception
Shenzhen Graduate School, Peking University
liumengyuan@pku.edu.cn

Hong Liu[†]
Key Laboratory of Machine Perception
Shenzhen Graduate School, Peking University
hongliu@pku.edu.cn

Chen Chen
Center for Research in Computer Vision
University of Central Florida
chenchen870713@gmail.com

Maryam Najafian
Center for Robust Speech Systems
University of Texas at Dallas
maryamnajafian@yahoo.com

Abstract

In order to efficiently recognize actions from depth sequences, we propose a novel feature, called Global Ternary Image (GTI), which implicitly encodes both motion regions and motion directions between consecutive depth frames via recording the changes of depth pixels. In this study, each pixel in GTI indicates one of the three possible states, namely positive, negative and neutral, which represents increased, decreased and same depth values, respectively. Since GTI is sensitive to the subject's speed, we obtain energy-based GTI (E-GTI) by extracting GTI from pairwise depth frames with equal motion energy. To involve temporal information among depth frames, we extract E-GTI using multiple settings of motion energy. Here, the noise can be effectively suppressed by describing E-GTIs using the Radon Transform (RT). The 3D action representation is formed as a result of feeding the hierarchical combination of RTs to the Bag of Visual Words model (BoVW). From the extensive experiments on four benchmark datasets, namely MSRAction3D, DHA, MSRGesture3D and SKIG, it is evident that the hierarchical E-GTI outperforms the existing methods in 3D action recognition. We tested our proposed approach on extended MSRAction3D dataset to further investigate and verify its robustness against partial occlusions, noise and speed.

1. Introduction

Recognizing human actions in the real-world environment finds applications in a variety of domains including intelligent video surveillance, customer attributes, and shopping behavior analysis. Common approaches in action recognition (e.g. hug, hand wave, smoke, etc.) rely on color cameras to record actions as RGB sequences and developed distinctive action representations to improve recognition accuracy [1, 2, 3].

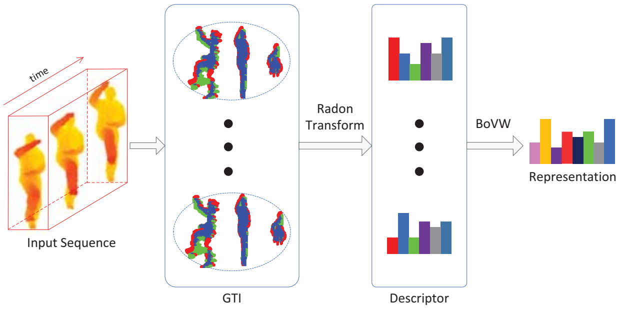


Figure 1: **The pipeline of extracting proposed representation.**

Accurate recognition of actions is a highly challenging task due to cluttered backgrounds, occlusions, light intensity and viewpoint variations [1]. In fact, extracting and encoding 3D motions from depth sequences that contain redundant data and background noise is often cited as an issue for the 3D action recognition [4, 5]. With rapid advances of imaging technology in capturing depth information in real time, there has been a growing interest in solving action recognition problems by using depth data (using depth cameras such as the Microsoft Kinect RGB-D camera). Depth data is estimated by infrared radiation and it is robust against the changes in lighting conditions compared to the conventional RGB data [6]. In addition to that, subtracting foreground from cluttered background can be done more accurately using the depth information which is independent from nuisance background attributes such as texture and color [7]. Moreover, accurate three-dimensional information of the subjects/objects can be exploited from the pre-

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.14

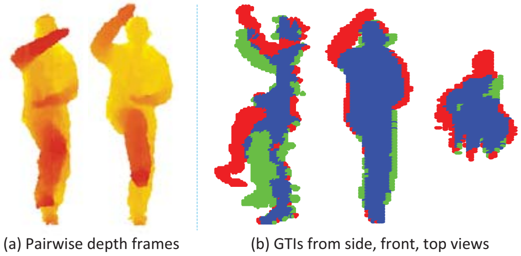

(a) Pairwise depth frames
(b) GTIs from side, front, top views

Figure 2: **GTIs generated from side, front and top views.** In GTIs, the pixels in red, green and blue represent positive, negative and neutral states respectively.

cise depth maps recorded by the RGB-D cameras [8].

In this paper, we propose a compact and robust representation for 3D action recognition, based on a set of Global Ternary Images (GTIs). The idea behind using the GTI is motivated by the fact that 3D motions can be implicitly expressed by changes in the depth value from three orthogonal coordinates. As shown in Fig. 1, our proposed approach comprises of three main stages. First, we extract GTIs from consecutive depth frames by detecting the changes in the state of each depth pixel. Each pair of depth frames generates three GTIs, which reflect the motions from the side, front and top views. Second, we adopt Radon Transform (RT) for encoding the GTIs to eliminate noise and reduce redundant data. In the third stage, we treat motion descriptors for pairwise depth frames as basic "words" and apply Bag of Visual Words (BoVW) model [1, 9] to encode all descriptors by a single feature vector.

Unlike previous local feature-based approaches that ignore spatial information, GTI is a global feature, capable of capturing the spatial relationships among local parts of motion regions. An example of this is evident by comparing Fig. 2 (a) and Fig. 2 (b). Fig. 2 (a) shows that motions are not very noticeable across two original depth frames, while Fig. 2 (b) shows that motion regions and directions are more evident. This enhanced representation is derived as a result of presenting the 3D motions and pairwise depth frames in terms of three GTIs.

Even though GTI can properly encode motion information, it is highly sensitive to the speed changes. The same types of actions, performed at various speeds, lead to misalignments of depth frames, resulting in different numbers and appearances of GTIs. To solve this problem, we propose Energy-based GTI (E-GTI) by combining energy-based sampling method and GTI. Our sampling method addresses this problem by sampling a fixed number of frames with equal energy across consecutive frames. Fig. 3 captures the changes occurred when a subject moves from pose A to pose B at different speeds. Our observation is that when a subject changes from one pose to another, the energy cost to overcome gravity is independent of the speed of performing the action. Therefore, the energy cost between pose A and pose B in Fig. 3 (a) and Fig. 3 (b) are equal, i.e. $E_a = E_b$. When the energy for our sampling method is set to E_a or E_b, only pose A and pose B are sampled and the frames in the green boxes are ignored. In other words, the sampled frames are not related to the speed changes.

Since we use BoVW model to encode E-GTIs, the temporal dependencies among depth frames are ignored. To this end, the E-GTIs are extracted from depth sequences, using multiple settings of motion energy, and organize E-GTIs in a hierarchical structure. Here, E-GTIs that use high values of motion energy record dependencies among frames in a large scale, while those that use low values of motion energy record dependencies among frames in a small scale (nearby frames). Where temporal dependencies are involved, we exploit representation-level fusion to obtain 3D action representation.

It is worthwhile to highlight several properties of the proposed scheme. First, both motion regions and motion directions between consecutive depth frames are efficiently preserved using GTI. Second, E-GTI measure is independent of the subject's speed. Third, each E-GTI is described by RT, which ensures robustness of resulting descriptors against both partial occlusions and noises. Fourth, we incorporated temporal information among depth frames by extracting E-GTI from depth sequences in a hierarchical fashion. Finally, we achieved state-of-the-art results on four benchmark datasets; two of which are 3D action recognition datasets, namely MSRAction3D [10] and DHA [11], and the remaining are 3D gesture recognition datasets, namely MSRGesture3D [12] and SKIG [13]. The remainder of the paper is organized as follows. Section 2 reviews the related work. Sections 3 and 4 present the E-GTI and 3D action recognition framework using E-GTI, respectively. Section 5 describes the experimental description and analysis. Finally, Section 6 presents the conclusions.

2. Related Work

We review four types of methods on encoding 3D motions from depth sequences. As a traditional method for 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. [14] proposed a Motion Energy Image (MEI) to represent where motion has occurred in an RGB sequence. A Motion History Image (MHI) was also proposed, where each pixel's intensity in MHI is a function of temporal history information at that point. By incorporating an additional dimension of depth, Azary et al. [15] extended MHI to define a Motion Depth Surface (MDS), which captures most recent motions in the depth direction as well as within each frame. To make full use of depth information, Yang et al. [16] 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. Based on the concept of DMM, Chen et al. [17, 18] proposed an improved version, which stacks the depth values across an entire depth sequence for three orthogonal planes. DMMs-based representations, effectively transform the action recognition problem from 3D to 2D and they have been successfully applied to depth-based action recognition.

An alternative approach to encode motions in depth sequences is by describing action shapes. Li et al. [10] extracted points from the contours of planar projections of 3D depth maps and employed an action graph to model the distribution of sampled 3D points. Recently, Tang et al. [19] developed a Histogram of Oriented Normal Vectors (HONV) by concatenating histograms of zenith and azimuthal angles to capture local distribution of the orientation of an object surface. Oreifej et al. [20] extended HONV to 4D space with time, depth and spatial coordinates and presented a Histogram of Oriented 4D Normals (HON4D) descriptor to encode the surface normal orientation of human actions. HON4D, by jointly capturing the distribution of motion cues and dynamic shapes, exhibits more discriminative power than the approaches that separately encode the motion and shape information. To further increase the robustness of HON4D, Yang et al. [21] grouped local hypersurface normals into polynormals, and then aggregated low level polynormals into the Super Normal Vector (SNV).

Unlike the above three features, cloud points, which denote human actions as a cloud of local points, are suitable to tackle both partial occlusions and the noise of original depth data. Li *et al.* [10] 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. Vieira *et al.* [22] divided 3D points into similarly sized 4D grids and applied spatial-temporal occupancy patterns to encode them. Wang *et al.* [12] explored an extremely large sampling space of random occupancy pattern features and used a sparse coding method to encode those features.

3. Energy-based Global Ternary Image

Given the depth sequences that contain actions, the first step in 3D action recognition is to model 3D motions. To simplify the problem, we propose GTI to encode 3D motions on three projected views. Even though the data of original depth frames is significantly reduced by using 2D projected maps, GTI still proves to be more accurate in encoding the details of 3D motions. Since GTI is sensitive to speed, we further extend GTI to E-GTI, which is robust to speed changes. In this section, we present the details of the proposed E-GTI.

3.1. Global Binary Image

Frame difference method, which captures motion regions, has been used to describe motions between consecutive frames. Lu et. al. [23] used the frame difference method to capture motions between two frames from RGB sequences by extracting motion areas, where the RGB value in previous frame differs from that of the current frame. For depth sequences, Yang et al. [16] projected depth maps onto three views and constructed DMMs by simply stacking the motion regions across a whole sequence to describe 3D actions. As a mainstream feature for 3D action recognition, DMMs outperform other features by encoding 4D information of motion regions in three projected planes. DMMs significantly reduce the data of original depth sequence by using just three 2D maps. However, the motion information in DMMs is still ambiguous, because of overlapping, in distinguishing similar actions. We name the motion regions extracted from consecutive frames as Global Binary Images (GBIs), which show discriminative power in describing motions. Since DMMs are formed by stacking GBIs on three views, most of the information from GBIs is overlapped and is therefore not fully used. To solve this problem, we preserve all information from GBIs by using a BoVW model. Specifically, three GBIs are combined into a single feature for describing a pair of frames. All features extracted from a depth sequence are aggregated into a compact representation by using the BoVW model.

To take advantage of additional information on motion and shape from depth maps, we project each depth frame onto three orthogonal Cartesian planes. We discard background (i.e. zero) region and select the bounding box of foreground (i.e. non-zero) region on each projected map as the region of interest. To achieve scale invariance, the bounding boxes are normalized to their respective fixed sizes. This normalization successfully eliminates effects of different heights and motion extents of different subjects.

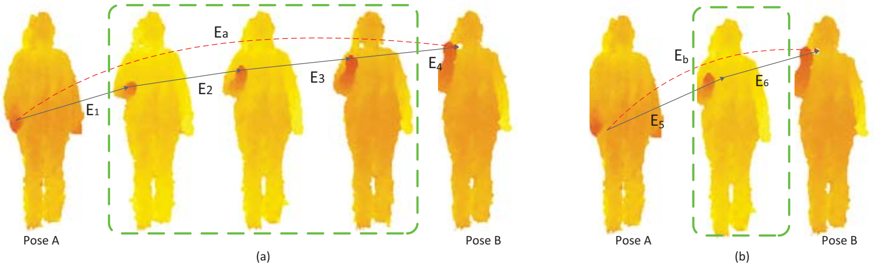


Figure 3: **Illustration of energy cost between pairwise poses.** (a) One person puts up the right hand at a slow speed. (b) One person puts up the right hand at a fast speed. The frames in the green boxes are generated by different speeds. E_a and E_b mean energy between pairwise frames. E_1, E_2, E_3, E_4, E_5, E_6 mean the energy between consecutive frames. We estimate the energy between pairwise frames by accumulating the energy between consecutive frames, e.g. $E_a = E_1 + E_2 + E_3 + E_4$, $E_b = E_5 + E_6$.

After these steps, the i-th depth map from a depth sequence can generate three 2D maps on the front, side, and top views respectively (i.e. map_f^i, map_s^i, map_t^i). For each view, we compute and threshold the differences between consecutive maps and obtain binary maps to indicate motion regions. In the following discussion, each binary map is called a GBI. Compared to the original depth maps, GBIs are less sensitive to noise, and therefore, they can provide more accurate clues regarding action-related motions.

Here, a 3D action is represented by an input depth sequence $\mathcal{I}$, where I^i represents the i-th frame, and the maximum number of frames is represented by N.

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

For every i-th frame, we estimate the inter-frame motion between I^{i-1} and I^i. The depth value for each pixel of consecutive frames (I^{i-1} and I^i) exhibits two useful properties. First, the 3D shape information of actors can be reflected 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, we project I^{i-1} and I^i onto three orthogonal planes

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

Cluttered backgrounds usually introduce ambiguity and reduce the accuracy of foreground action recognition. To determine the scope of foreground, we accumulate projected maps across a whole sequence and then find the bounding box of the accumulated projected maps. The bounding box is the smallest rectangle, which contains all the regions that an actor can ever reach. Foregrounds are extracted from projected maps, by using bounding boxes, and then resized to fixed sizes. Inter-frame motions are calculated from pairwise foregrounds, and applying size normalization, leads to robustness against scale variability across all inter-frame motions.

A GBI is a binary map, which can be defined as the motion region between two consecutive maps,

$$GBI_v^i = |map_v^i - map_v^{i-1}| > T_r, \tag{3}$$

where i ranges from 2 to N, and T_r is a threshold. We determine T_r by Otsu's method, an image thresholding algorithm, which maximizes the separability of the resultant classes in gray levels [24]. The definition of GBI in Eq. 3 implies that the motion regions generated by any kind of motions are reflected in the GBI. Horizontal motions, shared by multiple types of actions, directly introduce ambiguity in distinguishing similar actions. Therefore, this approach may not appeal to real applications where the actions are taking place during the subject's movement. To solve this problem, we propose an improved GBIs, which is less affected by variations in horizontal motions.

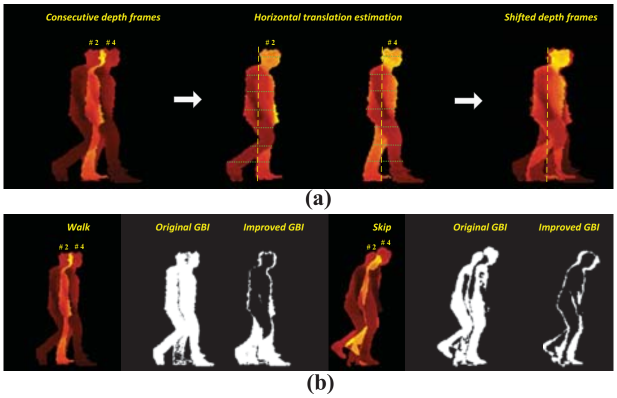


Figure 4: **Original GBIs and improved GBIs:** (a) A reference line is assigned to each depth frame, and by shifting the reference lines to the same position the horizontal translation between consecutive frames is compensated; (b) Given two actions, i.e. walking and skipping the original GBIs from depth frames and the improved GBIs from shifted depth frames show that the improved GBIs have better discriminative power compared to the original GBIs for distinguishing between similar actions.

Here, we first estimate horizontal motions between two consecutive maps, and then compensate them by shifting maps. Finally, GBIs are calculated by using the original definition. To estimate horizontal motions, we define a vertical line for each map, and then assume that the horizontal distance between consecutive vertical lines represents the horizontal shift. Indeed, when the actions are performed while the subject are still the position of each vertical line needs to be stable. To get the position of vertical line, we calculate the average position of foreground pixels on each line and then average these positions across all the lines. The comparison between original GBIs and the improved GBIs is illustrated in Fig. 4. It is to be pointed out here that the improved GBIs also are referred to as GBIs in the following sections.

3.2. Global Ternary Image

In addition to the motion regions captured by the GBIs, motion directions are also necessary for describing motions. For this reason, we assign directional information to GBIs. In RGB sequences, optical flow [25] is a benchmark method to calculate frame-by-frame motions, where both motion regions and motion directions are captured. Here, the performance of optical flow deteriorates as a result of lack of textural information. For extracting optical flow, the motion field needs to be nearly smooth, while the depth values often change dramatically during the process. This leads to another major problem and limits the application of this technique. As a result, we have to obtain motion directions from depth data by detecting changes in depth values.

For example, we determine the motions for red, green and blue points, as shown in Fig. 5 (c). For the blue point, the change in depth value is small; therefore, no motion is detected. This step is used to suppress the effect of noisy data and to maintain stability of motion areas. For the red

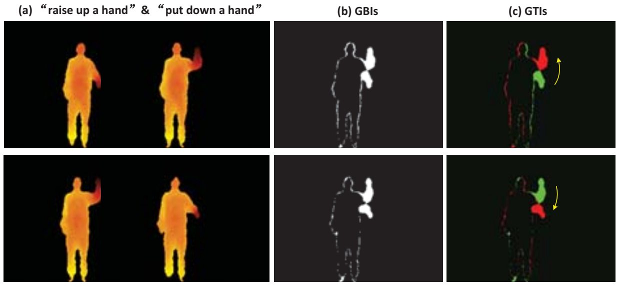


Figure 5: **Comparison between GBIs and GTIs.**

point, a big change in depth value is observed; therefore, motion could be detected at this point. The motion on the red point is considered positive, because the depth value increases from previous frame to current frame. For the same reason, the motion on the green point is considered negative. By repeating this procedure for all the points on the GBI, we can obtain a GTI which contains information regarding both motion region and motion direction.

Given GBI_v^i, we define the corresponding GTI by,

$$GTI_v^i = GBI_v^i \cdot \psi\{map_v^i, map_v^{i-1}\}, \tag{4}$$

where GBI_v^i and ψ respectively determine motion regions and motion directions, and the red/green color in GTI denotes motion directions. $\psi\{A, B\}$ is defined as,

$$\psi\{A, B\} = \begin{cases} 1, & if \;\; A > B \\ -1, & otherwise, \end{cases} \tag{5}$$

where each element of matrix A and the corresponding element of matrix B is compared.

To emphasize the merit of GTI, we compare it with GBI in Fig. 5, where two actions: "raise up a hand" and "put down a hand" are utilized for extracting both types of motion maps. Here, GBIs of these two actions look the same, but the GTIs look different. For instance, the information from GBI only suggests that the left hand of the subject moves in the vertical direction. While the information from GTI, suggests that the hand moves up for "raise up a hand" and moves down for "put down a hand".

3.3. Energy-based Global Ternary Image

The pace of each subject might vary under influence of different factors and time constraints. For example, the rate of change in the generated sequences might vary when an action is repeated several times by the same subject. This leads to increase in the inter-dissimilarities among the same type of actions. As for inter-frame motions, the speed of action directly affects the shape of motion. To tackle this problem, we convert the original depth sequences into speed-insensitive sequences, using energy-based sampling method. Based on the new speed-insensitive sequence, we extract GTIs from a sequence of frames, which results in a set of Energy-based GTI (E-GTI). In following parts, we focus mainly 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 $M$
Output: speed insensitive sequence $S_M$
for $v \in \{f, s, t\}$ do
    for $i = 2; i \leq N$ do
        $IMM_v^i \leftarrow$ Eq. 3;
for $i = 2; i \leq N$ do
    $E^i \leftarrow$ Eq. 6;
$S^1 \leftarrow I^1, S^M \leftarrow I^N, m \leftarrow 1$;
while $m \leq M - 1$ do
    for $i = 2; i \leq N - 1$ do
        if $(\frac{E^N}{M-1} \cdot m) \leq E^i$ then
            $S^{m+1} \leftarrow I^i$;
            $m \leftarrow m + 1$;
return $S_{M+1} = \{S^m\}_{m=1}^M$;
```

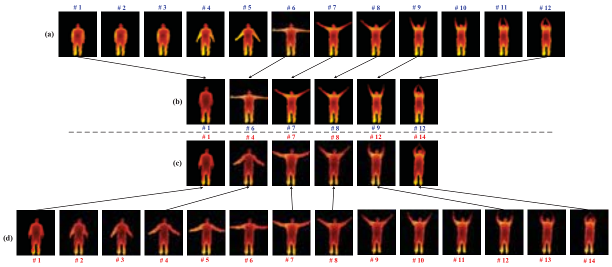


Figure 6: **Energy-based sampling method:** (a) and (d) Original depth sequences performing same type of action with different speeds; (b) and (c) Speed-insensitive sequences extracted by selecting frames from original sequences.

Given a depth sequence with N frames, we define the accumulated motion energy on the i-th frame thus,

$$E^i = \sum\nolimits_{v \in \{f,s,t\}} \sum\nolimits_{j=2}^{i} sum\{GBI_v^i\}. \tag{6}$$

Here, $sum\{\cdot\}$ returns the number of non-zero elements in a binary map. As observed in [21], the accumulated motion energy on a frame reflects the relative motion status (relates to the whole sequence). Different from [26], the energy here stands for motion rather than temperature. In Algorithm 1, we select frames from a given depth sequence to construct a speed-insensitive sequence with M frames. This pipeline can be divided into two steps. Initially, the first and the last frames are selected as the starting and ending frames of final sequence.Then, $M-2$ frames are selected to make sure that the motion energies between consecutive frames are nearly equal. As shown in Figs. 6 (a) and (d), the action of "waving two hands" is performed with different speeds. Following Algorithm 1, we obtain the new sequences in Figs. 6 (b) and (d), where the parameter M is set to six. It can be seen that the inter-dissimilarities between (b) and (c) are much smaller than those between (a) and (d), which indicates that the new sequences are speed-insensitive.

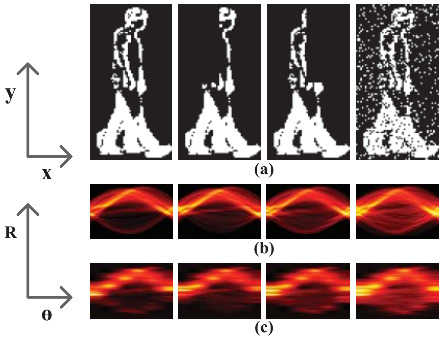


Figure 7: **GBIs and their corresponding Radon Transform:** (a) GBI is generated from depth frames under various conditions like partial occlusions or noises; (b) Radon Transform with parameter θ, ranging from 0 to π at an interval of 1. (c) Radon Transform with parameter θ ranging from 0 to π at an interval of 30; for better observation, the resulting images in (c) are interpolated to be π in width.

In [21] motion energy has been exploited to adaptively divide a depth sequence into several segments with equal motion energy. Then, each segment is described and concatenated as an action representation. However, the effect of speed is not tackled in [21], since each segment is segmented directly from the original sequence. In this section, we show 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 between the two poses is a stable value, which is unrelated to speed. Therefore, we select frames from the original sequence and form a new sequence, in which the motion energies between consecutive frames are nearly the same. This leads to elimination of the effect of variations in speed in the new sequence. Based on the new sequence and following the same steps as those used for forming GTIs, E-GTIs are extracted. Here, E-GTIs inherit the merits of GTIs, while exhibiting robustness against the speed changes.

3.4. Radon Transform

Radon Transform in two dimensions [27] is the integral transform consisting of the integral of a function over straight lines. In other words, Radon Transform can find the projection of a shape on any given line. Given a compactly supported continuous function $f(x, y)$ on $\mathbb{R}^2$, the Radon Transform is defined as:

$$\begin{aligned}\mathscr{R}\{f(x,y),\theta\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\cdot \\ \delta(xcos\theta + ysin\theta - \rho)dxdyd\rho,\end{aligned} \tag{7}$$

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 $[\lfloor -\frac{\sqrt{W^2+H^2}}{2} \rfloor, \lceil \frac{\sqrt{W^2+H^2}}{2} \rceil]$.

Chen et al. [28] used Radon Transform to describe human postures. Inspired by their work, we propose a GBIs based approach using Radon Transform and show that the robustness of Radon Transform to partial occlusions and noisy data is beneficial in describing the GBIs. For example, several GBIs in Fig. 7 (a) are described by Radon Transform. Despite the occlusions or noise in GBIs, their corresponding Radon Transforms in Fig. 7 (b) remain nearly the same. Let GBI_v^i denote a GBI, then the corresponding Radon Transform R_v^i can be formulated as:

$$R_v^i = \Big[\mathscr{R}\{GBI_v^i, \theta_j\}\Big]_{j=1}^{J}, \tag{8}$$

where $\theta_j \in [0 \;\; \pi]$. As shown in Figs. 7 (b) and (c), we set θ_j to range from 0 to π at an interval of respectively one and 30. Obviously, the images in (c) can preserve the general structural information of the images in (b), but contain less data than that of the images in (b). Based on this observation, we set θ_j to different values for Radon Transform, resulting in a low dimension, yet informative, descriptor for GBIs. Suppose GTI_v^i denotes a GTI, we convert it into two GBIs: $+GTI_v^i \cdot (GTI_v^i > 0)$ and $-GTI_v^i \cdot (GTI_v^i < 0)$. We use Eq. 8 to describe each GBI and concatenate both the results as the descriptor for GTI.

4. 3D Action Recognition

4.1. GTI based representation

Bag of Visual Words (BoVW) model is one of the popular methods for obtaining a compact representation from local features. Based on this model, a depth sequence can be represented as a bag of GTIs, where three GTIs of front, side, and top views of two consecutive depth maps are combined to form a feature. Let R_v^i denote the Radon Transform of GTI^i from view v, where $v \in \{f, s, t\}$. Then, GTI^i can be described thus:

$$R^i = \{\, R_f^i,\;\; R_s^i,\;\; R_t^i \,\}. \tag{9}$$

For a sequence $\mathcal{I}$ with N frames, we construct a feature set $R = \{R^i\}_{i=2}^N$ by extracting GTIs from the second frame to the N-th frame. During the training stage of BoVW, we randomly select local features from the training set and cluster them into K "words" using any clustering method, such as K-means [29]. During the testing stage, BoVW model finds the corresponding "word" for each feature in the feature set R and then uses the histogram of "words" as an effective representation of $\mathcal{I}$:

$$\begin{aligned}B_{GTI} &= \mathcal{B}\{R, K\} \\ &= \mathcal{B}\{\{R^i\}_{i=2}^N, K\}.\end{aligned} \tag{10}$$

To remove the effect of the number of local features, we further normalize the above representation as follows:

$$B_{GTI} = \frac{\mathcal{B}\{\{R^i\}_{i=2}^N, K\}}{||\mathcal{B}\{\{R^i\}_{i=2}^N, K\}||_1}, \tag{11}$$

where $||\cdot||_1$ calculates the l_1 norm.

Table 1: Results with different parameters.

Accuracy (%)	$P = 2$	$P = 4$	$P = 6$	$P = 8$
$K = 500$	91.43	94.99	95.41	94.08
$K = 1000$	91.69	95.30	**95.70**	95.00
$K = 1500$	92.93	95.08	94.66	94.34

4.2. Hierarchical E-GTI

Suppose a depth sequence $\mathcal{I}$ with N frames can be converted to a speed-insensitive sequence S_M with M frames. It is not appropriate to directly use S_M in describing $\mathcal{I}$, because most motion information from $\mathcal{I}$ is ignored in S_M. To compensate for the motion information, we extract multiple speed-insensitive sequences to give a detailed description of $\mathcal{I}$. To achieve this, we set parameter M in Algorithm 1 to $S_{M_1}, ..., S_{M_L}$, which produces L speed-insensitive sequences for one original sequence.

In Section 4.1, we denote B_{GTI} as the original depth sequence $\mathcal{I}$. Similarly, we let $B_{E-GTI}^{S_{M_L}}$ denote the representation for S_{M_L}. By concatenating all representations for speed-insensitive sequences, we obtain the following representation-level fusion representation:

$$B_{E-GTI}^{\mathcal{I}} = \left[B_{E-GTI}^{S_{M_1}}, \ \cdots, \ B_{E-GTI}^{S_{M_L}} \right], \qquad (12)$$

which can implicitly capture the temporal information of the original sequence $\mathcal{I}$.

5. Experiments

MSRAction3D dataset, introduced in [10], contains 20 actions, performed two or three times by ten subjects facing the depth camera, resulting in 567 depth sequences. DHA dataset, proposed in [11], contains 23 action categories, performed by 21 people, resulting in 483 depth sequences. MSRGesture3D dataset, proposed in [12], is a hand-gesture dataset. It contains 12 gestures, performed two or three times by each subject, resulting in 336 depth sequences. SKIG dataset, proposed in [13], contains 1080 hand-gesture depth sequences. It contains 10 gestures, performed with hand (i.e., fist, flat and index) by 6 subjects under two different illumination conditions and three backgrounds.

The recognition is conducted using a non-linear SVM with a homogeneous Chi2 kernel [30] and the 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 from vlfeat library[1]. To ensure that the results reported are consistent with those of other works, we adopt the same cross-validation methods as those given in [10], [11], [12] and [13]. It is to be noted that cross-subject validation is adopted for MSRAction3D dataset, with subjects $\#1, 3, 5, 7, 9$ for training and subjects $\#2,\ 4,\ 6,\ 8,\ 10$ for testing [10].

Table 2: Evaluation of GBI and GTI

Accuracy(%)	B_{GBI}	B_{GTI}
MSRAction3D	90.33%	95.70%
MSRAction3D-Order	38.63%	71.37%

Table 4: Comparison of our method and the previous approaches on four benchmark datasets. The word "Best" means the best published results so far.

3D Action Dataset	3D Action		3D Gesture	
	MSRAction3D	DHA	MSRGesture3D	SKIG
Original	74.70% [10]	86.80% [11]	88.50% [12]	88.70% [13]
Best	95.62% [31]	95.00%[32]	96.23% [33]	93.80% [34]
Bag of GTIs	95.70%	91.92%	96.42%	90.87%
Hierarchical E-GTIs	**97.22%**	**95.44%**	**98.80%**	**93.88%**

Table 5: Evaluation of the robustness to partial occlusions

3D Action Dataset	ROP [12]	ROP+SC [12]	n-RT	RT
MSRAction3D	-	-	88.20%	**95.70%**
MSRAction3D-Occlusion1	83.05%	86.17%	73.16%	**90.55%**
MSRAction3D-Occlusion2	84.18%	86.50%	51.33%	**93.45%**
MSRAction3D-Occlusion3	78.76%	80.09%	80.16%	**91.3%**
MSRAction3D-Occlusion4	82.12%	85.49%	79.87%	**88.61%**
MSRAction3D-Occlusion5	84.48%	**87.51%**	62.08%	87.39%
MSRAction3D-Occlusion6	82.46%	87.51%	71.66%	**89.07%**
MSRAction3D-Occlusion7	80.10%	83.80%	70.89%	**91.52%**
MSRAction3D-Occlusion8	85.83%	86.83%	80.56%	**94.13%**

To test the effect of Radon Transform and GTI, we use B_{GTI}, which is generated by performing Bag of GTIs model on original depth sequences, as the baseline representation. Let K be the number of clusters for K-means and P be the number of projections for Radon Transform. When P equals p, we conclude that θ_j in Eq. 8 equals $pi/p, pi/p*2, ...pi/p*p$. To select proper K and P, we test the effect on recognition accuracy of one parameter and keep the other parameter with default value. As shown in Table. 1, K and P change respectively from 500 to 1500 at an interval of 500, and from 2 to 8 at an interval of 2. We set default values of parameters K and P as 1000 and 6, using which we obtain the highest accuracy of 95.70%.

5.1. GBI & GTI

To verify the significance of encoding directions, we compare B_{GTI} and B_{GBI} on a new dataset, named MSRAction3D-Order. To create this dataset, for each action sequence in the original MSRAction3D dataset, we invert its temporal order (e.g., the first frame becomes the last frame) to generate a new sequence. As a results, the number of action sequences in MSRAction3D-Order dataset is doubled, i.e., $567 \times 2 = 1134$. Table 2 shows the performances of B_{GTI} and B_{GBI} on MSRAction3D and MSRAction3D-Order datasets. As expected, B_{GBI} performs worse on MSRAction3D-Order dataset than on MSRAction3D dataset. This is because one action type and its opposite type share similarities in motion shapes, which bring extra challenges to classification. Unlikely, B_{GTI} achieves accuracy of 71.37%, which is higher than that of B_{GBI}. This improvement is justifiable because GTI captures directional information, which is essential to distinguish one action type from its opposite type.

[1] Non-linear SVM classifier is implemented in http://www.vlfeat.org/applications/caltech-101-code.html

Table 3: Evaluation of hierarchical structure. The second column named “0” refers to applying BoVW model on original sequence.

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**

5.2. Hierarchical E-GTI

We convert original sequences into speed-insensitive sequences with $10,\ 20,\ 30,\ 40$ frames. We use B_{GTI}, $B_{GTI}^{S_{10}}$, $B_{GTI}^{S_{20}}$, $B_{GTI}^{S_{30}}$, $B_{GTI}^{S_{40}}$ (short for 0,1,2,3,4) (see Table 3) to describe the original sequence and the four corresponding speed insensitive sequences. Hierarchical E-GTI is denoted as $B_{GTI}^{\mathcal{I}}$, which is the representation-level fusion of $B_{GTI}^{S_{10}}$, $B_{GTI}^{S_{20}}$, $B_{GTI}^{S_{30}}$, $B_{GTI}^{S_{40}}$. Highest accuracies are observed with hierarchical E-GTI. Our method is compared with related works in Table 4, where we achieve better results on all datasets than the state-of-the-art methods.

5.3. Evaluation of Robustness

Robustness to partial occlusion: The partial occlusions are simulated using MSRAction3D dataset as described in [12]. Each volume of the depth sequence is divided into two parts along y, x and t coordinates, resulting in eight subvolumes. The occlusion is simulated by ignoring the depth data in one of the subvolumes. In Table 5, RT achieves an accuracy of around 90%, which outperforms Random Occupancy Pattern (ROP) and “ROP+sparse coding”. Further, we use a pixel value-based descriptor, instead of RT, for describing GTI. We refer to this method as “n-RT”, which concatenates all pixel values of GTI as a local feature. RT outperforms “n-RT”, because RT can suppress the effects of noisy data and partial occlusions (see Fig. 7).

Robustness to pepper noise: To simulate depth discontinuities in depth sequences, we add pepper noise in varying percentages (of the total number of image pixels) to depth images, as shown in Fig. 8 (a). Despite the effects of pepper noise, our method achieves more than 94% recognition accuracy on MSRAction3D dataset, as shown in Fig. 8 (b).

Robustness to speed: Based on MSRAction3D dataset, we construct an MSRAction3D-Speed dataset, by randomly selecting half the number of frames from testing sequences and then concatenating them to form new sequences. By comparing linear sampling method with our random sampling method (see Fig. 9), we conclude that action speeds in our new dataset change dramatically in a non-linear manner. We still achieve an accuracy of 87.17%, which demonstrates the robustness of our method to speed variations.

5.4. Time Cost

We compute 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 second on a 2.5GHz machine with 8GB RAM, using Matlab R2012a. The time for calculating the Radon Transform of a GTI is 0.0019 second. The overall computational time for calculating a proposed feature is thus about 0.0381 second.

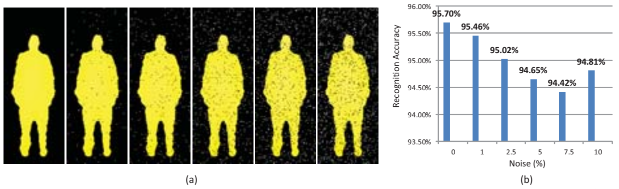
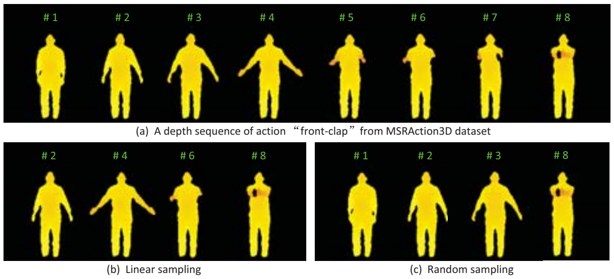


Figure 8: (a) Images affected by increasing percentages of pepper noise: 1%, 2.5%, 5%, 7.5% and 10%. (b) Recognition results on MSRAction3D dataset with different percentages of pepper noise.



(a) A depth sequence of action “front-clap” from MSRAction3D dataset

(b) Linear sampling

(c) Random sampling

Figure 9: Comparison between linear sampling and random sampling.

6. Conclusions

This paper presents a bag of GTI model to efficiently encode the information carried within motion regions and inter-frame motion directions. This model is robust against partial occlusions and depth discontinuities since the motion regions in GTIs are encoded by RT. To deal with varying speeds, we convert the original depth sequence into speed-insensitive sequences and propose a hierarchical GTIs framework to represent the action in the original sequence. As a result of using speed-insensitive sequences, our method is insensitive to the subject’s speed variations. Our method is extensively evaluated on four benchmark datasets designed for 3D action/gesture recognition, and achieves state-of-the-art results on these datasets.

7. Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC, No. 61340046, 60875050, 60675025, 61673030), National High Technology Research and Development Program of China (863 Program, No. 2006AA04Z247), 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. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008. 1, 2

[2] 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. 1

[3] I. Laptev, "On space-time interest points," *IJCV*, vol. 64, no. 2-3, pp. 107–123, 2005. 1

[4] M. Liu and H. Liu, "Depth Context: A new descriptor for human activity recognition by using sole depth sequences," *Neurocomputing*, pp. 747–758, 2015. 1

[5] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of Real-Time Image Processing*, pp. 1–9, 2013. 1

[6] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012. 1

[7] 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. 1

[8] B. Ni, G. Wang, and P. Moulin, "Rgbd-hudaact: A color-depth video database for human daily activity recognition," in *Advances in Computer Vision & Pattern Recognition*, pp. 193–208, 2013. 2

[9] M. Liu, H. Liu, and Q. Sun, "Action classification by exploring directional co-occurrence of weighted STIPs," in *ICIP*, 2014. 2

[10] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *CVPRW*, pp. 9–14, 2010. 2, 3, 7

[11] 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. 2, 7

[12] 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. 2, 3, 7, 8

[13] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *IJCAI*, pp. 1493–1500, 2013. 2, 7

[14] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *TPAMI*, vol. 23, no. 3, pp. 257–267, 2001. 2

[15] S. Azary and A. Savakis, "Grassmannian sparse representations and motion depth surfaces for 3D action recognition," in *CVPRW*, pp. 492–499, 2013. 2

[16] X. Yang, C. Zhang, and Y. L. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," *ACM MM*, pp. 1057–1060, 2012. 2, 3

[17] 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. 3

[18] 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*, pp. 3331–3337, 2016. 3

[19] 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. 3

[20] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences," in *CVPR*, pp. 716–723, 2013. 3

[21] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *CVPR*, pp. 804–811, 2014. 3, 5, 6

[22] 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. 3

[23] G. Lu and M. Kudo, "Learning action patterns in difference images for efficient action recognition," *Neurocomputing*, vol. 123, pp. 328–336, 2014. 3

[24] N. Otsu, "A threshold selection method from gray-level histograms," *TSMC*, vol. 9, no. 1, pp. 62–66, 1979. 4

[25] B. K. P. Horn and B. G. Schunck, "Determining optical flow: a retrospective," *AI*, vol. 59, no. 93, pp. 81–87, 1993. 4

[26] H. Zhang, A. Hedge, and B. Guo, "Surface and indoor temperature effects on user thermal responses to holding a simulated tablet computer," *Journal of Electronic Packaging*, vol. 138, 2016. 5

[27] S. R. Deans, "Applications of the radon transform," *Wiley Interseience Publications*, 1983. 6

[28] Y. Chen, Q. Wu, and X. He, "Human action recognition based on radon transform," *Studies in Computational Intelligence*, vol. 346, pp. 369–389, 2011. 6

[29] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *ACM MM*, pp. 1469–1472, 2010. 6

[30] A. Zisserman and Oxford, "Efficient additive kernels via explicit feature maps," *TPAMI*, pp. 480–492, 2012. 7

[31] C. Lu, J. Jia, and C. K. Tang, "Range-Sample depth feature for action recognition," in *CVPR*, pp. 772–779, 2014. 7

[32] 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. 7

[33] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition," in *ECCV*, pp. 742–757, 2014. 7

[34] P. Cirujeda and X. Binefa, "4DCov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences," in *3DV*, pp. 657–664, 2014. 7