

UDC 004.93

doi:10.31799/1684-8853-2020-5-2-11

## Human action recognition method based on conformal geometric algebra and recurrent neural network

Nguyen Nang Hung Van<sup>a</sup>, PhD Student, Specialist, [orcid.org/0000-0002-9963-7006](https://orcid.org/0000-0002-9963-7006), [nguyenvan@dut.udn.vn](mailto:nguyenvan@dut.udn.vn)

Pham Minh Tuan<sup>a</sup>, PhD, Lecturer, [orcid.org/0000-0001-9843-9676](https://orcid.org/0000-0001-9843-9676), [pmtuan@dut.udn.vn](mailto:pmtuan@dut.udn.vn)

Do Phuc Hao<sup>b</sup>, M. Sc., Lecturer, [orcid.org/0000-0003-0645-0021](https://orcid.org/0000-0003-0645-0021), [haodp@dau.edu.vn](mailto:haodp@dau.edu.vn)

Pham Cong Thang<sup>a</sup>, PhD, Lecturer, [orcid.org/0000-0002-6428-102X](https://orcid.org/0000-0002-6428-102X), [pcthang@dut.udn.vn](mailto:pcthang@dut.udn.vn)

Tachibana Kanta<sup>c</sup>, PhD, Associate Professor, [orcid.org/0000-0002-8675-7842](https://orcid.org/0000-0002-8675-7842), [kanta@cc.kogakuin.ac.jp](mailto:kanta@cc.kogakuin.ac.jp)

<sup>a</sup>The University of Danang – University of Science and Technology, Information Technology Faculty, 54 Nguyen Luong Bang St., Da Nang 550000, Vietnam

<sup>b</sup>Danang Architecture University, 566 Nui Thanh St., Da Nang 550000, Vietnam

<sup>c</sup>Kogakuin University, 1-24-2 Nishi-Shinjuku, Shinjuku-ku, Tokyo 163-8677, Japan

**Introduction:** Deep Learning plays an important role in machine learning and artificial intelligence. It is widely applied in many fields with high dimensional data, including natural language processing and image recognition. High dimensional data can lead to problems in machine learning, such as overfitting and degradation of accuracy. To address these issues, some methods were proposed to reduce dimensions of the data and computational complexity simultaneously. The drawback of these methods is that they only work well on data distributed on the plane. In the case of the data distributed on the hyper-sphere, such as objects moving in space, the processing results are not so good as expected. **Purpose:** The use of Conformal Geometric Algebra in order to extract features and simultaneously reduce the dimensionality of a dataset for human activity recognition using Recurrent Neural Network. **Results:** Human activity data in a 3-dimensional coordinate system is pre-processed and normalized by calculating deviations from the mean coordinate. Next, the data is transformed to vectors in Conformal Geometric Algebra space and its dimensions are reduced to return the feature vectors. Finally, we use the Recurrent Neural Network model to train feature vectors. Empirical results performed on the Motion Capture dataset with eight actions show that the Conformal Geometric Algebra combined with Recurrent Neural Network can give the best test results of 92.5%. **Practical relevance:** In human actions, some actions such as jump or dance will not move in motion and other actions, such as run, walk, will move in space. Therefore, we need a method to standardize actions. In the case of the data distributed on the hyper-sphere, the developed method can help us to extract features and simultaneously reduce the dimensionality of a dataset for human activity recognition using Recurrent Neural Network.

**Keywords** – activity recognition, Principal Components Analysis, Conformal Geometric Algebra, Deep Learning.

**For citation:** Nguyen Nang Hung Van, Pham Minh Tuan, Do Phuc Hao, Pham Cong Thang, Tachibana Kanta. Human action recognition method based on conformal geometric algebra and recurrent neural network. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 5, pp. 2–11. doi:10.31799/1684-8853-2020-5-2-11

### Introduction

Deep Learning (DL) is a new research trend in recent years for many applications, such as image processing, object detection, and remote control [1–4]. DL has two main models: Convolutional Neural Network (CNN) used to feature extraction in image processing [5, 6], and Recurrent Neural Network (RNN) used to handle sequence identification (sequence/time-series) [7].

The drawback of the Neural Network (NN) model is that each input  $x$  event is handled independently and gives the corresponding output  $y$  without the exchange of information collected at each input  $x$  [8]. The RNN contains internal loops that are able to save the exchanged information and the saved information can be transferred from one step to another of NN. So RNN can be used to input data from image and video converted into sequences for recognition or prediction problems. However, the most challenging problem of DL is still the selection process of data preprocessing and feature extraction techniques for training models.

Some commonly used machine learning models, such as Principal Components Analysis (PCA) [9, 10], Principal Components Regression (PCR) [11], and Multi-class Linear Discriminant Analysis (MLDA) [12], were proposed to reduce dimensions of the data and computational complexity simultaneously for training models. These machine learning method only work well with data distributed on a plane, such as face recognition or image classification [13]. In the case of data distributed on hyper-sphere, eg. moving objects in space, it is difficult to calculate accurately with the above methods. To address this issue, in this paper, we propose to use the Conformal Geometric Algebra (CGA) to extract features and reduce the number of data dimensions during the training of the RNN models.

In recent years, there have been a number of studies that have successfully applied Geometric Algebra (GA) for dimensionality reduction in some applications, such as color image processing, signal processing, and time-Series analysis [14–16]. CGA is part of GA, and a vector in CGA space is called a

conformal vector. Each of conformal vector is represented for  $m + 2$  dimensions of hyper-plane and hyper-spheres (see [17–20] for more details).

In this work, we propose to use principal components in  $m + 2$  dimensional CGA space. The feature vectors are gotten by eliminating the less relative components. We propose to transfer data from real space  $\mathbf{R}^m$  to conformal vectors as a set of points in CGA space  $\mathbf{P} \in \mathcal{G}_{m+2,1}$ . The selection of principal components determines the eigenvalues and eigenvectors. The eigenvectors of  $\mathbf{A}$  (conformal vector) are arranged in descending order. Then,  $k$  smallest eigenvectors are removed to reduce the data dimensions. It leads to receive the conformal vectors  $\mathbf{B}$  with  $m - k$  main components. These conformal vectors  $\mathbf{B}$  are used to train an RNN model.

## Related works

A growing interest in human action recognition using the DL model has recently arisen. To build a training model, we need to collect data via sensors or cameras [21, 22]. Next, it is important to use some preprocessing and machine learning methods for object feature extraction. Finally, we use the RNN model for action recognition.

Some methods are used in feature extraction for data dimensionality reduction such as PCA, LDA, and PCR. However, these methods are linear and it is hard to perform 3D relationships like linear motion or rotation. For example, a joint moves and rotates around the parent joint. Hence, motion data will be distributed on a sphere (or hyper-sphere) with the center coordinates of the parent joint.

Next, we present some feature extraction methods.

## Principal Components Analysis

Principal Components Analysis algorithm [9, 10] is usually used to convert dataset from a multi-dimensional space into a less dimensional space, but the method still ensures that the variance of the input data on each new dimension is the largest.

Given training set  $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{R}^d\}$ ,  $i = \{1, \dots, n\}$ ,

where  $\mathbf{x}_i$  is a vector in  $d$ -dimensional space, and  $n$  is the number of vectors in the set  $\mathbf{X}$ . PCA will perform a linear transformation to convert data into a new coordinate system. The linear transformation is defined by the scalar product of the vector  $\mathbf{x}$  and the unit vector of the weight  $\mathbf{w} \in \mathbf{R}^d$  where  $\|\mathbf{w}\| = 1$ . The problem is transformed into finding weight vectors so that the covariance of the linear transformation  $\mathbf{w}^T \mathbf{x}$  is the largest. We need to solve the problem:

$$\max_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^n (\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \boldsymbol{\mu})^2 \text{ s.t. } \|\mathbf{w}\|^2 = 1, \quad (1)$$

where the average of all vectors in the  $\mathbf{X}$  dataset is defined as follows:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i. \quad (2)$$

Equation (2) is the average of all vectors of the  $\mathbf{X}$  dataset. To solve this optimal problem, this paper introduces the Lagrange coefficient  $\lambda \geq 0$  for the Lagrange function as follows:

$$L(\mathbf{w}, \lambda) = \frac{1}{n} \sum_{i=1}^n (\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \boldsymbol{\mu})^2 - \lambda (\|\mathbf{w}\|^2 - 1). \quad (3)$$

Then, calculate the derivative of  $L(\mathbf{w}, \lambda)$  with  $\lambda$  going to zero will get the following formula:

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \mathbf{w} = \lambda \mathbf{w}. \quad (4)$$

So the optimization problem solved by decomposing Eigen is as follows:

$$\mathbf{C} \mathbf{w} = \lambda \mathbf{w}, \quad (5)$$

where

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T. \quad (6)$$

Equation (5),  $\mathbf{C}$  is the variance matrix of the  $\mathbf{X}$  data set. Finally, PCA uses a decrease in the number of dimensions of the data using the first  $k$  eigenvectors. These vectors are one that corresponds to the value of the maximum eigenvalues. This means that the original data set is approximated by data with less dimensionality and overview than the original data. The feature  $f(\mathbf{x})$  can be extracted from vector  $\mathbf{x}$  using the first  $k$  eigenvector as follows:

$$f_{\text{PCA}}(\mathbf{x}) = ((\mathbf{x} - \boldsymbol{\mu})^T \mathbf{w}_1, \dots, (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{w}_k)^T, \quad (7)$$

where  $\mathbf{w}_i$  is the  $i^{\text{th}}$  eigenvector,  $1 \leq i \leq k$ .

The extraction feature method using PCA uses a linear transformation for the input data. Hence, it is only possible to represent accurately the data distributed on the plane. However, the feature extraction results are not good with data distributed on the sphere. Furthermore, PCA mainly uses 3D coordinates during data collection, but it does not go in-depth into an analysis of 3D relationships of objects.

## Conformal Geometric Algebra

Given training set  $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{R}^d\}$ ,  $i \in \{1, \dots, n\}$  represented in real  $d$ -dimensional space. A CGA space is extended from the real Euclidean vector space  $\mathbf{R}^d$  by adding 2 orthonormal basis vector.

Thus, a CGA space is defined by  $d + 2$  basis vectors  $\{\mathbf{e}_1, \dots, \mathbf{e}_m, \mathbf{e}_+, \mathbf{e}_-\}$ , where  $\mathbf{e}_+$ ,  $\mathbf{e}_-$  and  $\mathbf{e}_i$ ,  $i \in \{1, \dots, d\}$  are defined as following:

$$\begin{aligned} \mathbf{e}_+^2 &= \mathbf{e}_+ \cdot \mathbf{e}_+ = 1; \\ \mathbf{e}_-^2 &= \mathbf{e}_- \cdot \mathbf{e}_- = -1; \\ \mathbf{e}_+ \cdot \mathbf{e}_- &= \mathbf{e}_+ \cdot \mathbf{e}_i = \mathbf{e}_- \cdot \mathbf{e}_i = 0, \forall i \in \{1, \dots, d\}. \end{aligned} \quad (8)$$

Thus, a CGA can be expressed by  $\mathcal{G}_{d+1,1}$ . This paper then defines the converted basis vectors  $\mathbf{e}_0$  and  $\mathbf{e}_\infty$  as

$$\mathbf{e}_0 = \frac{1}{2}(\mathbf{e}_- - \mathbf{e}_+), \quad \mathbf{e}_\infty = (\mathbf{e}_- + \mathbf{e}_+). \quad (9)$$

From Eq. (8) and (9), it is easy to see that:

$$\begin{aligned} \mathbf{e}_0 \cdot \mathbf{e}_0 &= \mathbf{e}_\infty \cdot \mathbf{e}_\infty = 0; \\ \mathbf{e}_0 \cdot \mathbf{e}_\infty &= \mathbf{e}_\infty \cdot \mathbf{e}_0 = -1; \\ \mathbf{e}_0 \cdot \mathbf{e}_i &= \mathbf{e}_\infty \cdot \mathbf{e}_i = 0, \forall i \in \{1, \dots, d\}. \end{aligned} \quad (10)$$

This training set is re-represented by the set of points  $\mathbf{P} \in \mathcal{G}_{d+1,1}$  in CGA space [23, 15] as follows:

$$\mathbf{P}_i = \mathbf{x}_i + \frac{1}{2}\|\mathbf{x}_i\|^2 \mathbf{e}_\infty + \mathbf{e}_0 \in \mathcal{G}_{d+1,1}. \quad (11)$$

Hence, a sphere in CGA space is represented as a conformal vector

$$\mathbf{S} = \mathbf{s} + s_\infty \mathbf{e}_\infty + s_0 \mathbf{e}_0. \quad (12)$$

The estimating process is performed by using the least squares  $d^2(\mathbf{P}_i, \mathbf{S})$ . The error function is defined as follows:

$$E = \sum_{i=1}^n d^2(\mathbf{P}_i, \mathbf{S}) = \sum_{i=1}^n \left( \mathbf{x}_i \mathbf{s}_j - s_{\infty j} - \frac{1}{2}\|\mathbf{x}_i\|^2 s_{0j} \right)^2. \quad (13)$$

This means that when minimizing the error  $E$  function,  $s$  can be limited by  $\|\mathbf{s}\|^2 = 1$ . In this case, the optimization problem becomes as follows:

$$\min \sum_{i=1}^n \left( \mathbf{x}_i \mathbf{s}_j - s_{\infty j} - \frac{1}{2}\|\mathbf{x}_i\|^2 s_{0j} \right)^2 \text{ s.t. } \|\mathbf{s}_j\|^2 = 1. \quad (14)$$

Following [14], the optimal problem is solved by the Eigen decomposition as follows:

$$\mathbf{A}_j \mathbf{s}_j = \lambda_j \mathbf{s}_j, \quad (15)$$

where  $\mathbf{A}$  is the variance matrix of the  $i^{\text{th}}$  training set in CGA space:

$$\mathbf{A} = \sum_{i=1}^n f(\mathbf{x}_i) f^T(\mathbf{x}_i). \quad (16)$$

The function  $f(\mathbf{x}_i)$  is defined as follows:

$$f(\mathbf{x}_i) = \mathbf{x} - f_\infty - \|\mathbf{x}\|^2 f_0 \in \mathbf{R}^m, \quad (17)$$

where

$$f_\infty = \frac{-\sum_4 \sum_{i=1}^n \mathbf{x}_i + \sum_2 \sum_{i=1}^n \|\mathbf{x}_i\|^2 \mathbf{x}_i}{\left(\sum_2\right)^2 - n \sum_4}; \quad (18)$$

$$f_0 = \frac{\sum_2 \sum_{i=1}^n \mathbf{x}_i - n \sum_{i=1}^n \|\mathbf{x}_i\|^2 \mathbf{x}_i}{\left(\sum_2\right)^2 - n \sum_4}, \quad (19)$$

$\sum_2 = \sum_{i=1}^n \|\mathbf{x}_i\|^2$  and the sum of the four powers  $\sum_4 = \sum_{i=1}^n \|\mathbf{x}_i\|^4$ .

An eigenvector  $\mathbf{s}_j$  is a conformal eigenvector of a subset  $\mathbf{X}_j$  defined in hyper-plane or hyper-sphere  $\mathbf{S}_j = \mathbf{s}_j + s_{\infty j} \mathbf{e}_\infty + s_{0j} \mathbf{e}_0$  and eigenvalues  $\lambda_j$  are variance;  $s$  and  $s_0$  are the scalar coefficient of the basis vectors  $\mathbf{e}$  and  $\mathbf{e}_0$ .

### Recurrent Neural Network

Recurrent Neural Network models [7] are mostly used in the fields of natural language processing, speech recognition, and action recognition. The learned model always has the same input size, because the input of each state is the output of the previous state. It allows to use same transition function with same parameters at every step. These properties make it possible to learn the model executing on all steps and all sequence lengths. Hence, RNN is able to generalize the sequence lengths not being in the training set. Therefore, the learned model will be estimated with much fewer data.

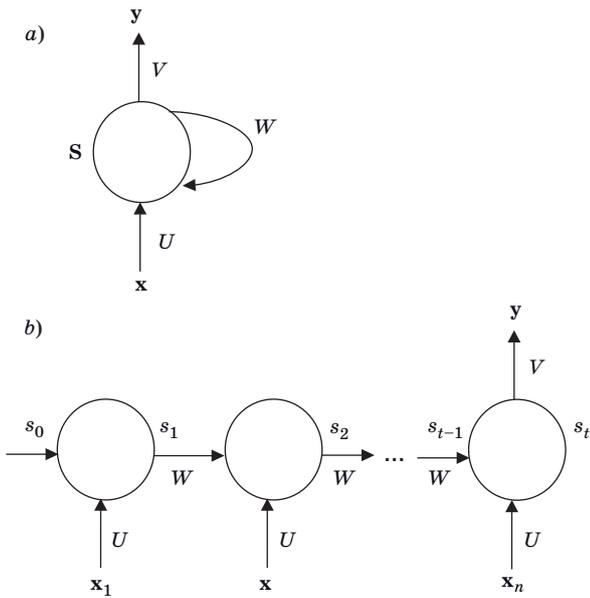
In this study, we will introduce RNN with data as human action videos (Fig. 1). These actions will go through PCA and CGA to create the feature vector  $\mathbf{x}$  of size  $n$ . The calculation will take many to one form of the RNN model, with multiple input and output predicting actions in the data set.

Each input value  $\mathbf{x}_t$  is of size  $n$ , after passing RNN, there will be an output value  $\mathbf{y}$  of size  $c$  (number of clustering), for each circle called a state, the input of each state is  $\mathbf{x}_t$  and  $s_{t-1}$  (which is the output of the previous state). Now output  $s_t$  is calculated in the following formula:

$$s_t = \tanh(U\mathbf{x}_t + Ws_{t-1}), \quad (20)$$

where  $s_i$  of size  $m$ ;  $U$  and  $W$  is hyperparameter [24]:  $U$  of size  $(m \times n)$  is a coefficient matrix between  $\mathbf{x}_{t-1}$  and  $\mathbf{x}_t$ ;  $W$  of size  $(m \times n)$  is a coefficient matrix connecting  $s_t$  and  $s_{t-1}$ . In Fig. 1, a  $V$  of size  $(c \times m)$  is a coefficient matrix converting  $\mathbf{x}_t$  to  $\mathbf{y}$ .

Because there is only one output value,  $\mathbf{y}$  can be determined through the activation function is *softmax*:



■ Fig. 1. Illustrate the many to one problem of the RNN model: a — short RNN model; b — RNN model for application

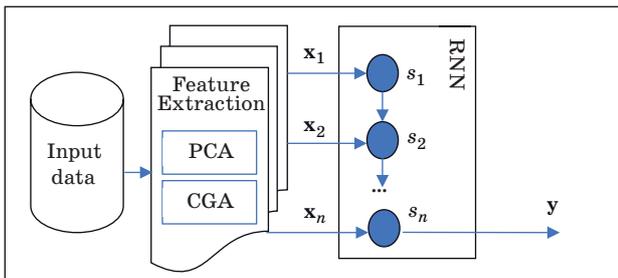
$$y = \text{softmax}(Vs_n). \quad (21)$$

From (21) from the input sequence  $\mathbf{x}$ , we also determine the output value quite simply.

Deep Learning has two big models that are CNN and RNN. CNN is a processing model in the problem of image input, and very common applications in computer vision are classification, object detection, and segmentation. However, in human action recognition, the input data of the models is different actions and the model needs to predict the appropriate output action (predict time series). Therefore, in this paper, we choose the many to one model in RNN to apply.

### The proposed method

The proposed method is to analyze data of moving objects and human actions with markers with



■ Fig. 2. The overview of proposed RNN model

coordinates in 3D space. Specifically, the proposed use of the CMU (Carnegie Mellon University) [25] motion capture dataset consists of 08 different actions, each action consisting of multiple files and each file consisting of corresponding frames. In each frame there are 41 markers (41 joints), each marker is each coordinates are represented in 3D space.

In this paper, we propose a technique to normalize data by moving the coordinate axes of all joints back to their original coordinates, then using PCA and CGA to extract features. Finally, use these feature vectors to create input values for the RNN (Fig. 2).

### Transformation method of coordinates

In human actions, some actions such as *jump* or *dance* will not move in motion and other actions such as *run*, *walk* will move in space. Therefore, we need a method to standardize actions to be similar. In this study, we propose to transform all markers to new coordinates by calculating deviations from the mean coordinate to extract features.

Give a data set:

$$\mathfrak{S} = \{\zeta_i | \zeta_i \in \mathbf{R}^{t(i) \times m \times 3}, i \in \{1, \dots, n\}, \quad (22)$$

where  $\zeta_i = [\delta_{i,1}^T, \dots, \delta_{i,t(i)}^T]^T \in \mathbf{R}^{t(i) \times m \times 3}$  is a vector of the  $i^{th}$  action;  $t(i)$  is frame number of the  $i^{th}$  action;  $\delta_{i,j} = [\theta_{i,j,1}^T, \dots, \theta_{i,j,m}^T]^T \in \mathbf{R}^{m \times 3}$  is a vector corresponding to the  $i^{th}$  frame of action and

$$\theta_{ijk} = \begin{bmatrix} \theta_{ijk1} \\ \theta_{ijk2} \\ \theta_{ijk3} \end{bmatrix} \in \mathbf{R}^3 \text{ is the coordinates of the } i^{th}$$

marker of the  $j^{th}$  frame of the  $i^{th}$  action.

Then, we convert action  $\zeta_i$  to the set of vectors as follows:

$$\mathbf{X} = \{\mathbf{x}_{ij} | \mathbf{x}_{ij} \in \mathbf{R}^{m \times 3}, j \in \{1, \dots, t(i)\}, \quad (23)$$

where  $\mathbf{x}_{ij}$  is a vector corresponding to features the  $j^{th}$  frame, and the  $i^{th}$  action:

$$\mathbf{x}_{ij} = [\mathbf{g}^T(\theta_{ij1}), \dots, \mathbf{g}^T(\theta_{ijk}), \dots, \mathbf{g}^T(\theta_{ijm})]^T, \quad (24)$$

where  $\mathbf{g}(\theta_{ijk})$  is a transformation of coordinates of the  $k^{th}$  marker of the  $j^{th}$  frame of the  $i^{th}$  action.

In this paper, we can use the function

$$\mathbf{g}^T(\theta_{ijk}) = \theta_{ijk} \quad (25)$$

or transformation method of coordinates

$$g(\theta_{jjk}) = \theta_{jjk} - \hat{\theta}_{ij}, \quad (26)$$

where  $\hat{\theta}_{ij} = \frac{1}{m} \sum_{k'=1}^m \theta_{jjk'}$ .

So, we have the new set of vector for the training set of PCA and CGA\_PCA.

### Model combining PCA with RNN

Recurrent Neural Network can combine well with other models to predict in time series (predict time series). However, RNN uses many parameters on each state, which can lead to over-fitting. By combining PCA with RNN, PCA is capable reduce feature dimensions so that network nodes can be reduced but retain the original properties (original information).

The PCA algorithm uses orthogonal transformations to convert the data set from a multi-dimensional space to a new space with less dimension. This transformation is based on finding the axis of the new space so that the method of data projected on that axis is greatest. From Eq. (22) and (23), we have  $\mathbf{X}_{new}$  as follows:

$$\mathbf{X} = \{\mathbf{x}_{ij} | \mathbf{x}_{ij} \in \mathbf{R}^{m \times 3}; j \in \{1, \dots, t(i)\}, i \in \{1, \dots, n\}\}. \quad (27)$$

Now, we need to solve the problem:

$$\max_{\mathbf{w}} \frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} (\mathbf{w}^T \mathbf{x}_{ij} - \mathbf{w}^T \boldsymbol{\mu})^2 \quad s.t \|\mathbf{w}\|^2 = 1, \quad (28)$$

where

$$\boldsymbol{\mu} = \frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} \mathbf{x}_{ij}. \quad (29)$$

Equation (28), to the optimal problem, this paper introduces the Lagrange coefficient  $\lambda \geq 0$  for the Lagrange function as follows:

$$L(\mathbf{w}, \lambda) = \frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} (\mathbf{w}^T \mathbf{x}_{ij} - \mathbf{w}^T \boldsymbol{\mu})^2 - \lambda (\|\mathbf{w}\|^2 - 1). \quad (30)$$

Then, calculate the derivative of  $L(\mathbf{w}, \lambda)$  with  $\lambda$  going to zero will get the following formula:

$$\frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} (\mathbf{x}_{ij} - \boldsymbol{\mu})(\mathbf{x}_{ij} - \boldsymbol{\mu})^T \mathbf{w} = \lambda \mathbf{w}. \quad (31)$$

Then, the eigenvalues can be obtained via the following function:

$$\mathbf{C}\mathbf{w} = \lambda \mathbf{w}, \quad (32)$$

where

$$\mathbf{C} = \frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} (\mathbf{x}_{ij} - \boldsymbol{\mu})(\mathbf{x}_{ij} - \boldsymbol{\mu})^T. \quad (33)$$

Equation (32),  $\mathbf{C}$  is the variance matrix of the  $\mathbf{X}$  data set. Finally, PCA uses a decrease in the number of dimensions of the data using the first  $k$  eigenvectors. The feature  $f_{PCA}(\mathbf{x})$  can be extracted from vector  $\mathbf{x}$  using the first  $k$  eigenvector as follows:

$$f_{PCA}(\mathbf{x}) = ((\mathbf{x} - \boldsymbol{\mu})^T \mathbf{w}_1, \dots, (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{w}_k)^T, \quad (34)$$

where  $\mathbf{w}_i$  is the  $i^{th}$  eigenvector,  $1 \leq i \leq k$ .

Now, we use the transform  $f_{PCA}(\mathbf{x})$  to apply the learning model by converting the data set  $\mathbf{T} = \{f_{PCA}(\mathbf{x}_{ij}), \mathbf{y}_i | \mathbf{x}_i \in \mathbf{R}^{m \times 3}, \mathbf{y}_i \in \{1, \dots, c\}\}, i \in \{1, \dots, n\}$ , where  $f_{PCA}(\mathbf{x}_{ij})$  and  $\mathbf{y}_i$  are label and feature vector after applying PCA.

Then, we use the data set  $\mathbf{T}$  to initialize the input data for the RNN model. From Eq. (20), the formula is rewritten as follows:

$$s_i = \tanh(U\mathbf{x}_{ij} + Ws_{i-1}). \quad (35)$$

Because there is only one output value,  $\mathbf{y}_i$  can be determined through the activation function is *softmax* and Eq. (21) is rewritten

$$\mathbf{y}_{PCA,i} = \text{softmax} \frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} (Vs_i). \quad (36)$$

From Eq. (36) we get the clustering result of each action. Because PCA uses linear methods and assumes that the data is distributed on the plane, in cases where the data is distributed in the hyper-sphere, the PCA will not give a high result. The study further suggests using CGA to combine with the RNN.

### Model combining CGA with RNN

This proposal will proceed to build the RNN model on CGA space. From Eq. (22) and (23) data is converted into points in CGA space as follows:

$$\mathbf{P}_{ij} = \mathbf{x}_{ij} + \frac{1}{2} \|\mathbf{x}_{ij}\|^2 \mathbf{e}_\infty + \mathbf{e}_0 \in \mathcal{G}_{m \times 3 + 1, 1}. \quad (37)$$

The process of estimating using least squares  $d^2(\mathbf{P}_i, \mathbf{S})$ . The error function  $E$  as follows:

$$E = \sum_{i=1}^n \sum_{j=1}^{t(i)} d^2(\mathbf{P}_{ij}, \mathbf{S}) = \sum_{i=1}^n \sum_{j=1}^{t(i)} \left( \mathbf{x}_{ij} \mathbf{S} - s_\infty - \frac{1}{2} \|\mathbf{x}_{ij}\|^2 s_0 \right)^2. \quad (38)$$

This means that when minimizing the error  $E$  function,  $\mathbf{s}$  can be limited by  $\|\mathbf{s}\|^2 = 1$ :

$$\min \sum_{i=1}^n \sum_{j=1}^{t(i)} \left( \mathbf{x}_{ij} \mathbf{s} - s_{\infty} - \frac{1}{2} \|\mathbf{x}_{ij}\|^2 s_0 \right)^2. \quad (39)$$

Therefore, we might be tempted to express the previous problem using a non-negative Lagrange multiplier  $\lambda$  as the minimization of

$$L(\mathbf{s}, \lambda) = \frac{1}{\sum_{i=1}^n \sum_{j=1}^{t(i)} t(i)} \sum_{i=1}^n \sum_{j=1}^{t(i)} \left( \mathbf{x}_{ij} \mathbf{s} - s_{\infty} - \frac{1}{2} \|\mathbf{x}_{ij}\|^2 s_0 \right)^2 - \lambda (\|\mathbf{s}\|^2 - 1). \quad (40)$$

The optimal result can be solved using Eigen problem

$$\mathbf{A} \mathbf{s} = \lambda \mathbf{s}, \quad (41)$$

where  $\mathbf{A}$  is the variance matrix of the  $i^{th}$  training set in CGA space:

$$\mathbf{A} = \sum_{i=1}^n \sum_{j=1}^{t(i)} f(\mathbf{x}_{ij}) f^T(\mathbf{x}_{ij}). \quad (42)$$

The function  $f_{CGA}(\mathbf{x}_i)$  is defined as follows:

$$f_{CGA}(\mathbf{x}_i) = \mathbf{x}_i - f_{\infty} - \|\mathbf{x}_i\|^2 f_0 \in \mathbf{R}^m, \quad (43)$$

where

$$f_{\infty} = \frac{-\sum_4 \sum_{i=1}^n \sum_{j=1}^{t(i)} \mathbf{x}_{ij} + \sum_2 \sum_{i=1}^n \sum_{j=1}^{t(i)} \|\mathbf{x}_{ij}\|^2 \mathbf{x}_{ij}}{(\sum_2)^2 - \sum_{i=1}^n t(i) \sum_4}; \quad (44)$$

$$f_0 = \frac{\sum_2 \sum_{i=1}^n \sum_{j=1}^{t(i)} \mathbf{x}_{ij} - \sum_{i=1}^n t(i) \sum_{i=1}^n t(i)}{(\sum_2)^2 - \sum_{i=1}^n t(i) \sum_4}. \quad (45)$$

Similar to the PCA model, RNN uses input data is  $f_{CGA}(\mathbf{x}_i)$  after using the CGA to extraction feature. From Eq. (20), (21) and (43) can be rewritten as

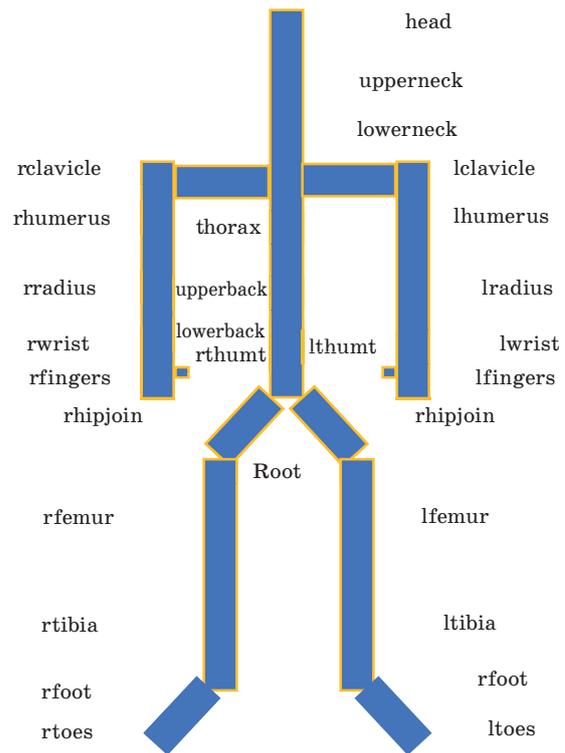
$$y_{CGA,i} = \text{softmax} \frac{1}{\sum_{i=1}^n t(i)} \sum_{i=1}^k \sum_{j=1}^{t(i)} (V s_i). \quad (46)$$

This model is implemented on CGA space, i. e. data in real space is transferred to CGA space. With the characteristics of CGA, it is possible to represent objects in space and geometric relationships very well. So movements with complex distributions like human joints use CGA very reasonably.

## Experimental

### Experimental data

The database of motions of CMU, USA [25] is free for all uses. Motions are captured in a working volume of approximately  $3 \times 8$  m. In this model, humans wear a black jumpsuit having 41 markers taped on (Fig. 3). The Vicon cameras see the markers in infrared. The images picked up from the various cameras are triangulated to get 3D data.



■ Fig. 3. Illustration model of markers on the body

■ Table 1. Database experiment

Action	Number frame		
	Training	Testing	Total
Dance	3.305	1.577	4.882
Jump	1.198	846	2.044
Kick	1.605	1.163	2.768
Placing Tee	1.487	1.096	2.583
Putt	1.534	974	2.508
Run	452	322	774
Swing	1.324	977	2.301
Walk	1.074	928	2.002
Total	11.979	7.883	19.862

This study uses 8 kinds of human action. They are *dancing, jumping, kicking, placing tea, putt, running, swing, walking*. This paper uses the warehouse file format .c3d and data include total frame is 19.869 frames, divide the number of frames of each action into 2 parts (*train — 60 % and the test — 40 %*). Details are presented in Table 1.

Figure 3 shows the labels of markers, which have 3D coordinates ( $x, y, z$ ).

**Predict with RNN**

This experiment is conducted with the original data set on the RNN model with the non-transformation using Eq. (25) and with the proposed transformation using Eq. (26). The parameters of the RNN network are the number of neural = 200, epochs = 20, and classes = 8 (8 kinds of human action), batch\_size = 5, and activation function is Tanh.

The results from Table 2 show that when the coordinates are moved closer to the original coordinates, the result is 84.45 %. This result is much higher than when keeping the coordinates is about 70.97 %. However, to improve as well as increase efficiency when using the RNN. In this study, we propose to use many methods of dimensional reduction before putting into the RNN to predict.

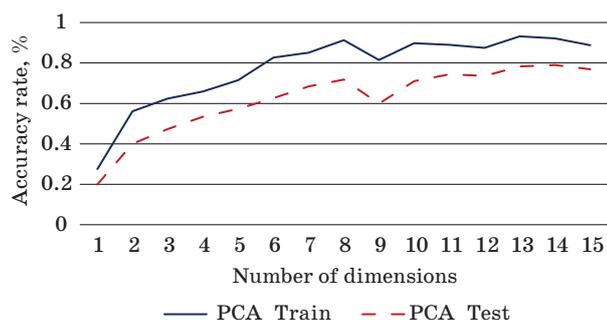
**Predict with PCA\_RNN**

In this experiment, we will conduct coordinate transformation before using PCA to extract features. Results Fig. 4 shows that as the number of dimensions increases, the result gradually increases.

When the number of dimensions is 13, the Train result is 93.12 % and the Test is 78.2 %, the result will not increase and can be considered to have converged at the number of dimensions equal to 14.

■ **Table 2.** Comparison results of two data sets using RNN, %

Part	non-Transformation	Transformation
Train	72.57	87.11
Test	70.97	84.45



■ **Fig. 4.** Results of combining PCA and RNN

Next, research should continue to experiment with the use of CGA to extract features.

**Predict with CGA\_RNN**

In this experiment, we will conduct preprocessing before using CGA to extract features. Fig. 5 shows that the result when using CGA will converge most when receiving the full attributes of the object. At the same time, the results clearly show that if you remove some key attributes, the result will decrease.

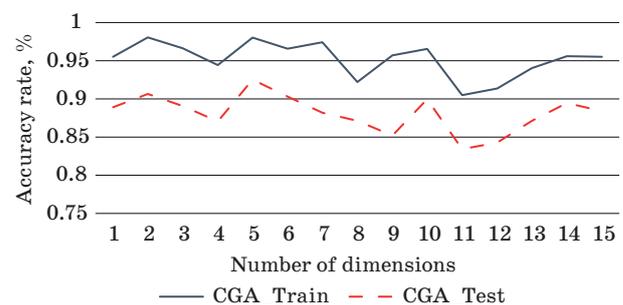
The highest train result was 98.1 % with a dimension is 2 and the highest test was 92.52 % with a dimension is 5.

**Evaluation of results**

The experiment was conducted with 5 times of implementation on the proposed method with the number of neural = 200, epochs = 20, the results achieved in Table 3. The results of using CGA\_RNN are much higher than PCA\_RNN, this is also true. when using CGA to represent moving objects in space.

In the previous study [26], we used PCA and CGA to extraction feature and predict on data set [25]. The best result is 88.86 %. However, when we use PCR to classify and predict the results, we encounter some limitations such as calculation speed and complexity. So we propose a method of combining CGA with RNN.

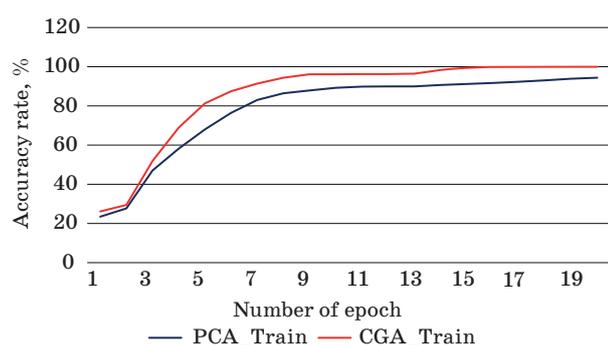
Currently, some studies are using ML models [27–29] and DL in human action recognition [5, 30, 31]. However, these studies only focus on developing RNN and Long Short Term Memory (LSTM) models to predict but do not care about the characteristics of the object and extraction feature methods



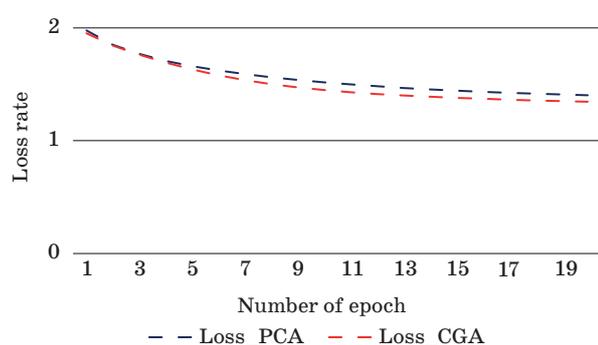
■ **Fig. 5.** Results of combining CGA and RNN

■ **Table 3.** Compares the results of the two proposed methods, %

Part	PCA_RNN	CGA_RNN
Train	85.24	95.59
Test	72.83	88.50



■ Fig. 6. Comparison of the results of the two proposed methods



■ Fig. 7. The loss rate of the two methods with epoch = 20

of the data (object). In the study [32], the authors used Android phones to collect actionable data and then used the LSTM-RNN model to train and the research results were quite high at 96 %. Although the number of markers (joints) is only 3, this study has the number of markers is 41.

The results of Figs. 4 and 5, the maximum results are achieved by the PCA method with the number of dimension 13, and by the CGA method with the number of dimension 2.

Figure 6 shows that as the number of epochs increases, the recognition accuracy will increase.

When epoch increases, the recognition results can also increase, and when epoch reaches a certain value, the results cannot increase (a straight line or descending).

Figure 7 shows the output loss in each step of RNN. The loss rate decreases, the recognition results increases. This figure also shows that the loss rate of the CGA method is lower than that of PCA during each epoch.

## Conclusion

In this paper, we proposed a normalization method for input data by calculating deviations from the mean coordinate, before using PCA and CGA to reduce the number of dimensions and create input data for the RNN network. Experimental results show that the proposed method CGA\_RNN has 88.50 % higher results than 72.83 % of PCA\_RNN. Theoretically, RNN can learn distant states. However, in reality, RNN only brings the previous states to the later stages.

However, the RNN model only carries a certain number of states after that, it would be vanishing gradients, and this model can only be learned from near states (short term memory). Therefore, it is necessary to apply the proposed model with LSTM to improve the research results. Researching CGA to apply to DL is a direction, creating a basis for analyzing large data of moving objects in space and other applications such as image processing, action recognition, and automatic control in the future.

## Acknowledgements

This work was supported by The University of Danang, University of Science and Technology, code number of Project T2019-02-20.

## References

1. Rosebrock A. *Deep Learning for Computer Vision*. PyimageSearch, 2017. 500 p.
2. Arthishwari K., Anand M. Design of LSTM-RNN on a sensor based HAR using Android phones. *International Journal of Recent Technology and Engineering (IJRTE)*, 2020, vol. 8, no. 5, pp. 2277–3878. doi:10.35940/ijrte.E6821.018520
3. Pham C. T., Kopylov A. Multi-quadratic dynamic programming procedure of edgepreserving denoising for medical images. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2015, XL-5/W6, pp. 101–106.
4. Goodfellow I., Bengio Y., and Courville Aaron. *Deep Learning*. MIT Press, 2018. 800 p. doi:10.1007/s10710-017-9314-z
5. Liu H., and Taniguchi T. Feature extraction and pattern recognition for human motion by a deep sparse autoencoder. *IEEE International Conference on Computer and Information Technology*, 2014, pp. 173–181. doi:10.1109/CIT.2014.144
6. Ji S., Xu W., Yang M., and Yu K. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, vol. 35, no. 1, pp. 221–231. doi:10.1109/TPAMI.2012.59
7. Du Y., Wang W., and Wang L. Hierarchical recurrent neural network for skeleton based action recognition.

- IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1110–1118.
8. Mikolov T., Bengio Y. and Pascanu R. On the difficulty of training recurrent neural networks. *International Conference on Machine Learning*, 2013, vol. 28, pp. 1310–1318. arXiv:1211.5063v2.
  9. Jolliffe I. T. *Principal Component Analysis*. 2nd Ed. New York, Springer-Verlag, 2002. 518 p.
  10. Smith L. I. *A tutorial on Principal Components Analysis*. Cornell University, USA, 2002. 27 p.
  11. Jolliffe I. T. A note on the use of principal components in regression. *Applied Statistics*, 1982, vol. 31, iss. 3, pp. 300–303.
  12. Alan J. I. *Linear Discriminant Analysis*. Springer, 2012. Pp. 1525–1548.
  13. Nixon M. S., and Aguado A. S. *Feature Extraction and Image Processing for Computer Vision*. 3rd Ed. ELSEVIER, 2012. 632 p.
  14. Pham M. T., Tachibana K., Hitzer E. M. S., Buchholz S., Yoshikawa T., and Furuhashi T. Feature Extractions with geometric algebra for classification of objects. *IEEE World Congress on Computational Intelligence*, Hongkong, 2008, pp.
  15. Hildenbrand D., and E. Hitzer. Analysis of point clouds using conformal geometric algebra. *3rd International Conference on Computer Graphics Theory and Applications*, Funchal, Madeira, Portugal, 2008, pp. 69–94.
  16. Minh Tuan Pham, Hao Do Phuc, Kanta Tachibana. Feature extraction for classification method using principal component based on conformal geometric algebra. *IEEE World Congress on Computational Intelligence / International Joint Conference on Neural Network*, 2016, no. 978-1-5090-0620, pp. 4112–4117.
  17. Hestenes D., and Sobczyk G. *Clifford algebra to geometric calculus. A unified language for mathematics and physics*. Springer, Dordrecht, 1984. 309 p. doi:10.1007/978-94-009-6292-7
  18. Doran C., and Lasenby A. *Geometric algebra for physicists*. Cambridge University Press, 2003. 578 p.
  19. Pham M. T., Tachibana K., Hitzer E. M. S., Yoshikawa T., and Furuhashi T. Classification and clustering of spatial patterns with geometric algebra. *AGACSE 2010*, Leipzig, 2010, no. 978-1-84996-107-3, pp. 231–247. doi:10.1007/978-1-84996-108-0\_12
  20. Pham M. T., Tachibana K., Yoshikawa T., and Furuhashi T. A clustering method for geometric data based on approximation using conformal geometric algebra. *2011 IEEE International Conference on Fuzzy Systems*, 2011, pp. 2540–2545. doi:10.1109/FUZZY.2011.6007574
  21. Hachaj T., Ogiela M. R., and Piekarczyk M. Dependence of Kinect sensors number and position on gestures recognition with Gesture Description Language semantic classifier. *Computer Science and Information Systems (FedCSIS), Federated Conference*, 2013, pp. 571–575.
  22. Afsar P., Cortez P., and Santos H. Automatic human action recognition from video using hidden Markov model. *IEEE 18th International Conference on Computational Science and Engineering*, 2015, pp. 105–109. doi.org/10.1109/CSE.2015.41
  23. Eckhard Hitzer, Tohru Nitta, and Yasuaki Kuroe. Applications of Clifford’s geometric algebra. *Advances in Applied Clifford Algebras*, 2013, vol. 23, iss. 2, pp. 377–404. arXiv:1305.5663v1.
  24. Goodfellow I., Bengio Y., and Courville A. *Sequence Modeling: Recurrent and Recursive Nets*. In: *Deep Learning*. MIT Press, 2016. Pp. 367–415.
  25. *The Carnegie Mellon University, The Carnegie Mellon University Motion Capture Database*. Available at: <http://mocap.cs.cmu.edu> (accessed 20 April 2020).
  26. Nguyen N. H. V., Pham M. T., Do P. H. Marker selection for human activity recognition using combination of conformal geometric algebra and principal component regression. *Proceedings of the Seventh International Symposium on Information and Communication Technology*, December 8–9, 2016, pp. 274–379. doi:10.1145/3011077.3011133
  27. Md. Al Mehedi Hasan, Omar Faruqe. Face recognition using PCA and SVM. *Anti-counterfeiting, Security, and Identification in Communication, ASID 2009*, pp. 97–101.
  28. Gehrig D., and Schultz T. Selecting relevant features for human motion recognition. *19th International Conference on Pattern Recognition*, Tampa, FL., 2008, pp. 1–4. doi:10.1109/ICPR.2008.4761290
  29. K. G. Manosha Chathuramali, Ranga Rodrigo. Faster human activity recognition with SVM. *The International Conference on Advances in ICT for Emerging Regions*, Colombo, 2012, pp. 197–203.
  30. Wei L., and Shah S. K. Human activity recognition using deep neural network with contextual information. *The 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pp. 34–43.
  31. da Silva R. E., Ondrej J., and Smolic A. Using LSTM for automatic classification of human motion capture data. *14th International Conference on Computer Graphics Theory and Applications, VISIGRAPP 2019*, pp. 236–243.
  32. Arthishwari K., Anand M. Design of LSTM-RNN on a sensor based HAR using Android phones. *International Journal of Recent Technology and Engineering (IJRTE)*, 2020, vol. 8, no. 5, pp. 2277–3878.

УДК 004.93

doi:10.31799/1684-8853-2020-5-2-11

**Метод распознавания действий человека на основе конформной геометрической алгебры и рекуррентной нейронной сети**Нгуен Нанг Хунг Ван<sup>а</sup>, аспирант, Specialist, orcid.org/0000-0002-9963-7006, nguyenvan@dut.udn.vnФам Минь Туан<sup>а</sup>, PhD, преподаватель, orcid.org/0000-0001-9843-9676, pmtuan@dut.udn.vnДо Фук Хао<sup>б</sup>, M. Sc., преподаватель, orcid.org/0000-0003-0645-0021, haodp@dau.edu.vnФам Конг Тханг<sup>а</sup>, PhD, преподаватель, orcid.org/0000-0002-6428-102X, pcthang@dut.udn.vnТатибана Канта<sup>в</sup>, PhD, доцент, orcid.org/0000-0002-8675-7842, kanta@cc.kogakuin.ac.jp<sup>а</sup>Университет Дананга — Университет науки и технологий, факультет информационных технологий, ул. Гнунг Лыонг Банг 54, Дананг, 550000, Вьетнам<sup>б</sup>Данангский университет архитектуры, ул. Нуи Тхань 566, Дананг, 550000, Вьетнам<sup>в</sup>Университет Когакуин, Ниси-Синдзюку 1-24-2, Синдзюку-ку, Токио, 163-8677, Япония

**Постановка проблемы:** глубокое обучение играет важную роль в машинном обучении и искусственном интеллекте. Оно находит широкое применение во многих областях, где приходится оперировать большими массивами данных, например в обработке естественных языков или распознавании изображений. Высокая размерность данных ведет к таким проблемам машинного обучения, как чрезмерное обучение или падение точности. Для их преодоления предлагаются методы одновременного снижения размерности данных и вычислительной сложности. Недостаток этих методов состоит в их ориентированности на данные, распределенные по плоскости. В случае данных, распределенных по гиперсфере, таких как передвигающийся в пространстве объект, результаты обработки показывают качество ниже ожидаемого. **Цель:** применение конформной геометрической алгебры для одновременного снижения размерности массива данных, необходимых для выделения признаков и распознавания действий человека с использованием рекуррентной нейронной сети. **Результаты:** данные о действиях человека в трехмерной системе координат подвергаются предобработке и нормализации путем вычисления отклонений от средних координат. Далее данные преобразуются в векторы в пространстве конформной геометрической алгебры, а их размерность снижается для извлечения векторов признаков. Наконец, применяется модель рекуррентной нейронной сети для обучения векторов признаков. Опытные результаты, полученные на массиве данных захвата движений для восьми действий, показали, что комбинация конформной геометрической алгебры с рекуррентной нейронной сетью обеспечивает наилучший результат в 92,5 % случаев. **Практическая значимость:** некоторые действия человека, например прыжок или танец, не сопряжены с перемещением в пространстве, в отличие от таких, как бег или ходьба. Поэтому необходим способ стандартизации действий. В случае данных, распределенных по гиперсфере, разработанный метод позволяет выделять признаки с одновременным снижением размерности массива данных для распознавания действий человека посредством рекуррентной нейронной сети.

**Ключевые слова** — распознавание действий, анализ принципиальных компонент, конформная геометрическая алгебра, глубокое обучение.

**Для цитирования:** Nguyen Nang Hung Van, Pham Minh Tuan, Do Phuc Hao, Pham Cong Thang, Tachibana Kanta. Human action recognition method based on conformal geometric algebra and recurrent neural network. *Информационно-управляющие системы*, 2020, № 5, с. 2–11. doi:10.31799/1684-8853-2020-5-2-11

**For citation:** Nguyen Nang Hung Van, Pham Minh Tuan, Do Phuc Hao, Pham Cong Thang, Tachibana Kanta. Human action recognition method based on conformal geometric algebra and recurrent neural network. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 5, pp. 2–11. doi:10.31799/1684-8853-2020-5-2-11

В статье Burkov A. A., Shneer S. V., Turlikov A. M. Lower bound for average delay in unblocked random access algorithm with orthogonal preambles, 2020, № 3, на с. 84 в разделе Acknowledgment вместо project number 19-31-27001

следует читать

project number 19-37-90041.