IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 4, 2021, accepted February 19, 2021, date of publication March 2, 2021, date of current version March 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063302

A Body Part Embedding Model With Datasets for Measuring 2D Human Motion Similarity

JONGHYUK PARK[1,2], SUKHYUN CHO[1,2], DONGWOO KIM[3], OLEKSANDR BAILO[3], HEEWOONG PARK[1,2], SANGHOON HONG[3], AND JONGHUN PARK[1,2]

[1]Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea
[2]Center for Superintelligence, Institute for Industrial Systems Innovation, Seoul National University, Seoul 08826, Republic of Korea
[3]Kakao Brain, Seongnam 13494, Republic of Korea

Corresponding author: Jonghun Park (jonghun@snu.ac.kr)

This work was supported in part by Kakao and Kakao Brain corporations, and in part by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science and ICT (MSIT) under Grant NRF-2019R1F1A1053366.

ABSTRACT Human motion similarity is practiced in many fields, including action recognition, anomaly detection, and human performance evaluation. While many computer vision tasks have benefited from deep learning, measuring motion similarity has attracted less attention, particularly due to the lack of large datasets. To address this problem, we introduce two datasets: a synthetic motion dataset for model training and a dataset containing human annotations of real-world video clip pairs for motion similarity evaluation. Furthermore, in order to compute the motion similarity from these datasets, we propose a deep learning model that produces motion embeddings suitable for measuring the similarity between different motions of each human body part. The network is trained with the proposed motion variation loss to robustly distinguish even subtly different motions. The proposed approach outperforms the other baselines considered in terms of correlations between motion similarity predictions and human annotations while being suitable for real-time action analysis. Both datasets and codes are released to the public.

INDEX TERMS Computer vision, dataset, deep learning, human pose, metric learning, motion similarity.

I. INTRODUCTION

Human motion, essentially a combination of translation and rotational motions of each body joint, contains a lot of information inherent to a human. In particular, motion similarity that can be obtained by analyzing the human motions has a wide range of applications. For instance, the motion similarity can be used for action recognition [1]–[6]. It is also possible to measure a motion similarity to determine whether a task is performed well [7]–[11] or to identify abnormal behavior [12]–[14]. A motion comparison system is helpful for matching a target person from different cameras for re-identification [12], [14]–[18].

While analyzing human motion plays an essential role in the tasks mentioned above, motion similarity research has attracted less attention so far due to the following reasons. First, measuring the motion similarity is a challenging problem. Different camera views or human body structures cause a variety of 2D joint coordinates, even for similar motions in videos. This makes it impossible to directly measure the similarity using the joint coordinates. Second, the availability of large-scale datasets for learning the motion similarity is limited. Lastly, to the best of our knowledge, there are few human motion datasets available for assessing the performance of different motion similarity computation methods.

This work attempts to compare short video clips of basic human motions. Our target motions are short enough (1−2 seconds) to be described by a few words or a sentence and can be easily imitated after a demonstration. The comparison is solely made by body movements, excluding the difference in body sizes and appearances. In this work, we represent a motion as a sequence of joint positions and do not consider interactions between a human and an object in the environment. To build a comparison system, we extend the model of [19] to split human motion into five body parts and map them to a latent space. The similarity is measured by comparing the encoded motion vectors. The overview of the method for measuring the motion similarity is depicted in Fig. 1.

The proposed model is trained on our synthetic motion dataset, an extended version of the dataset [19], from Adobe

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Napoletano.

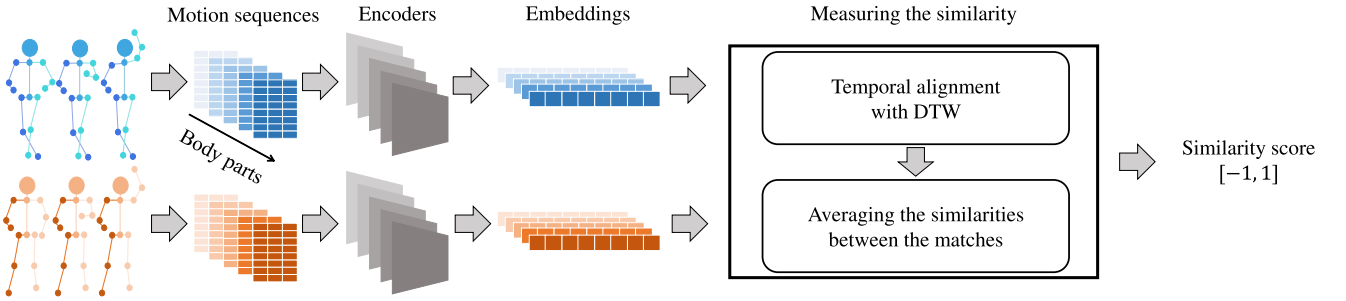


FIGURE 1. A high-level overview of the proposed method. The model takes a sequence of human joints coordinates and produces embeddings of body parts which are used to analyze the similarity between different motions.

Mixamo [20]. We have collected human motion animations with variations in characteristic elements (e.g., movement and angle) of each motion. These variations are important for learning the motion similarity as they allow us to distinguish very similar movements.

To effectively incorporate this property into the model, we propose a motion variation loss. This loss enforces the distance between two motion embeddings to be proportional to the motion variation. Overall, our goal is to learn disentangled motion embeddings from the skeletons and camera views, in contrast to the existing motion similarity learning methods utilized in other tasks such as action recognition. The motion embeddings, which are divided into five body parts and learned through the motion variation loss in this paper, allow the robust analysis of the motion similarity.

To assess the performance of the proposed model on real-world data, we have utilized NTU RGB+D 120 [21], [22] dataset that has been widely used for action recognition [23]–[31]. NTU RGB+D 120 is composed of videos where people perform various actions with different camera angles. Since there are no labels in the dataset to measure the similarity, we have collected labels via Amazon Mechanical Turk (AMT) [32]. The proposed method achieved the highest correlations between the evaluated similarity scores and the human perceptions compared to the other baseline models considered. Both datasets and codes are made publicly available.[1]

In summary, our main contributions are:

- Body part embedding (BPE) model that can measure motion similarity and identify movement differences of body parts.
- Motion variation loss to distinguish subtle variations in similar motions.
- Synthetic dataset for training the intra-motion variations and human annotations of NTU RGB+D 120 for motion similarity evaluation.

The remainder of this paper is organized as follows. First, we offer a briefing on the related work in Section II. The training method and algorithm for calculating motion similarity are described in Section III. In Section IV, we introduce the datasets used for the model training and validation. Section V provides results of the experiments and an application example. Finally, the last section summarizes and concludes the paper.

[1]https://chico2121.github.io/bpe/

II. RELATED WORK

A. MOTION SIMILARITY

Defining a similarity between human poses is a fundamental task for building a video retrieval system, and many studies have approached it in various ways. Ferrari *et al.* [33] defined a feature vector representing a human pose in an image based on the pictorial structure, computed vector space distances between poses in a query image and individual frames of a video in a database, and aggregated the distances from all frames to obtain the relevant score for the video track. The system in [34] measured the distance between poses as a function of joint angles and retrieved similar videos containing the frames near a query pose image. Kim and Kim [9] measured the similarity between two dance poses using the joint angles of the person in a frame. In [35], fixed-length short motion sequences are clustered into groups and used as a motion representation. In contrast to [9], [33], [34], in which the methods for either image-level or video-level retrieval rely on pose similarity between two image frames, our method defines motion similarity between a pair of motion sequences directly.

Apart from analyzing independent motions, Shen *et al.* [36] proposed an approach for measuring motion similarity in interaction-based activities. While this approach is promising for the tasks of interacting with objects, it cannot be utilized for comparing independent movements without interactions or when the objects are located far apart. Moreover, the algorithm is unsuitable for real-time applications. In [4], Long Short Term Memory (LSTM) with a layer normalization architecture was utilized to generate motion embeddings. At the training phase, the authors replaced hard-negative mining required for similarity learning with Maximum Mean Discrepancy (MMD) [37] and saved computational cost. We employ this architecture as one of the main baselines when assessing the performance of our model.

B. HUMAN BODY EMBEDDING

Decomposing a body into several parts based on the human skeleton structure and constructing representations for individual parts are common approaches for human

action understanding. For instance, Choutas *et al.* [38] suggested a fixed-size representation of a video clip containing a motion as a collection of trajectory maps of individual joints. Guo and Choi [39] argued that learning local representations separately on four limbs and torso was helpful for short-term human motion prediction.

Liu *et al.* [40] suggested the hierarchical partwise bag-of-words (HPBoW) representation focused on visual salience of different body areas with 7 bag-of-words features (limb, head, leg, foot, upper, lower, and full) in 3 levels (low level, middle level, and high level). Hake [41] extracted interaction triples – a body part, an action verb, and an object – from images based on the features of part regions. Jammalamadaka *et al.* [42] proposed a method to classify a body part image to a corresponding class and constructed an image embedding vector based on the classification scores. In [42], the learned body part embeddings are not merely intermediate representations for subsequent classifiers, but also contain general information for 2D pose reconstruction and can be appropriate for measuring motion similarity.

C. DATASETS FOR MEASURING THE SIMILARITY

Few datasets contain a pair of motions with similarity annotation. Mori *et al.* [43] suggested annotating pose similarity automatically by determining whether the mean joint distance satisfies a given threshold constraint. Despite the ease of constructing a large-scale dataset, this method was not able to generate similar pairs of poses in terms of human perception. Other studies, [35], [36], evaluated their models against binary classification or retrieval test sets constructed from action recognition datasets by regarding motions of the same action label as being similar. In [36], the authors defined an evaluation task in which a comparison system is required to assign higher similarities to motions that share more specific class labels for a query motion. However, this class-based strategy cannot properly capture the intra-class variations of motions as actions from the same class might be less similar than another pair from different classes.

Motivated by the lack of motion similarity datasets, we propose a new dataset containing motion similarity annotations obtained from crowd workers for about 20K video pairs.

D. TRIPLET AND QUADRUPLET LOSSES

Schroff *et al.* [44] proposed a triplet loss which takes three images as input. Specifically, the input is composed of a reference image of a person (anchor), another image of the same person (positive sample), and an image of a different person (negative sample). The loss minimizes the distance of anchor-positive features while maximizing the distance of anchor-negative features. The triplet loss has been actively applied in many studies. Wohlhart and Lepetit [45] utilized it to predict classes of objects and 3D poses. Hermans *et al.* [46] proposed a triplet loss that includes a sampling method, showing state-of-the-art performance in person re-identification. Kim *et al.* [47] proposed a new triplet loss using continuous labels that preserve the distance ratio of numeric labels in the learned latent space, allowing the model to learn the degree of similarity, not just the order. Meanwhile, [48], [49] learned the distance between features using four samples. By using the triplet loss as a basis, the authors constrain the minimum inter-class distance to be larger than the maximum intra-class distance. While such approaches focused on the inter-class separation through a manually defined constraint, we aim to map the distance in a latent space between the intra-class samples using ground-truth motion variation labels.

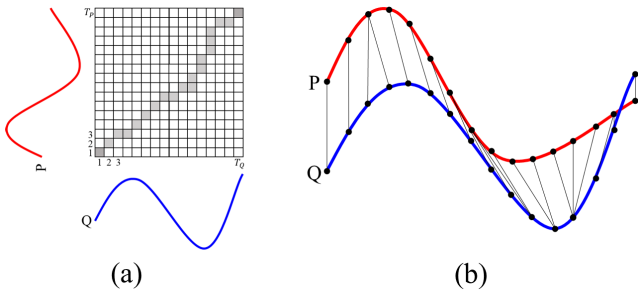


FIGURE 2. Visualization of DTW algorithm. (a) is a figure describing the process of finding the optimal alignment between two time series *P* and *Q*. (b) shows that the elements of the time series *P* and *Q* were matched according to the optimal alignment obtained by DTW.

E. DYNAMIC TIME WARPING

Dynamic Time Warping (DTW) [50] is an algorithm that determines an optimal alignment of two time series with different lengths. Let $P = (p_1, p_2, \cdots, p_{T_P}) \in \mathbb{R}^{h_p \times T_P}$ denote the time series of h_p-dimensional vectors with time-length T_P. Similar to P, $Q = (q_1, q_2, \cdots, q_{T_Q}) \in \mathbb{R}^{h_q \times T_Q}$ represents the time series of h_q-dimensional vectors with time-length T_Q. To align the two time series, DTW constructs cost matrix $D \in \mathbb{R}^{T_P \times T_Q}$ using dynamic programming. Each matrix element $D_{ij} = d(p_i, q_j)$ is the cost between p_i and q_j, where $i \in [1 : T_P]$, $j \in [1 : T_Q]$, and $d(\cdot)$ is a distance metric. The optimal alignment is the path with the smallest sum of costs from D_{11} to $D_{T_P T_Q}$, like the gray colored path in Fig. 2 (a). The path obtained in this way is not matched with the same time points, but with the points of a similar pattern, as shown in Fig. 2 (b). We utilize DTW to align two motions and calculate their similarity.

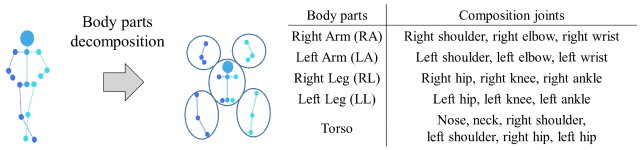
III. METHOD

We propose a learning-based method to encode unique motion embeddings necessary for the motion similarity assessment. Inspired by the framework of [19], we extend the method by training a model to reconstruct each body part (Section III-A1), rather than the whole body, to identify particular hand or foot movements. Furthermore, we propose a motion variation loss (Section III-A2) to robustly calculate motion similarity (Section III-B).

A. BODY PART EMBEDDING MODEL

1) NETWORK STRUCTURE

Let $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{C}$ respectively denote the sets of motion, skeleton, and camera view attributes in the training set. To calculate



Body parts	Composition joints
Right Arm (RA)	Right shoulder, right elbow, right wrist
Left Arm (LA)	Left shoulder, left elbow, left wrist
Right Leg (RL)	Right hip, right knee, right ankle
Left Leg (LL)	Left hip, left knee, left ankle
Torso	Nose, neck, right shoulder, left shoulder, right hip, left hip

FIGURE 3. Body parts decomposition. Middle hip is the origin of the coordinate system.

a total loss, we require $M = \{m, m', m''\}$, $S = \{s, s'\}$, and $C = \{c, c'\}$, which are subsets of $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{C}$ respectively. Also, m and m'' are required to be from the same motion class with different characteristics (i.e., motion variation), and m' to be a motion from a different class. For example, if m is a *Low jump*, then m'' is a *High jump* whereas m' is a *Sitting*. s and s' represent two skeletons with different body structures, and c and c' are view angles when 3D motion is projected to 2D. We can generate a motion sequence by selecting and combining each element from M, S, and C. Let $\mathcal{X} = \{X_{ijk} \in \mathbb{R}^{2\times J\times T} \mid i \in M, j \in S, k \in C\}$ be the set of 2D coordinate sequences where J is the number of joints of a skeleton, and T is the time length of the motion sequence. Among the elements of $\mathcal{X}$, X_{msc} and $X_{ms'c}$ are the sequences of 2D joint coordinates representing the same motion m with the different skeletons s and s' at the same view angle c.

With set B, composed of $n_B = 5$ body parts, we decompose a skeleton to construct body part embeddings. In our case, $B = \{Right\ Arm, Left\ Arm, Right\ Leg, Left\ Leg, Torso\}$ is considered, as depicted in Fig. 3. Specifically, the motion sequence X_{msc} is decomposed into specific body parts $X^b_{msc} \in \mathbb{R}^{2\times n_b\times T}$, where n_b is the number of joints in $b \in B$. X^b_{msc} is fed into body part motion encoder E^b_M and skeleton encoder E^b_S to produce embeddings. For global camera view encoder E_C, all the decomposed motion sequences are concatenated to generate the input of E_C. Let us denote this input as $X^a_{msc} \in \mathbb{R}^{2\times n_c\times T}$ where n_c is the sum of the number of joints that make up each body part. The embeddings from the two types of encoders E^b_M and E^b_S, respectively capturing the motion and skeleton features of body part b, are combined with the feature from the E_C. These combined features are decoded by body part decoder D^b to reconstruct the body part motion sequence. Since we consider 5 body parts, each of the motion and skeleton encoders has 5 modules (i.e., one for each body part) that do not share weights among them. This process is visualized in Fig. 4.

2) LOSSES

Aberman *et al.* [19] used a triplet loss to enforce separation between the samples on the motion latent space. Let $z^b_{msc} = E^b_M(X^b_{msc})$ be the resulting motion embedding of X^b_{msc} obtained from E^b_M. Then, motion triplet loss is:

$$\mathcal{L}^b_M(X^b_{msc}, X^b_{m's'c'}) = [d(z^b_{ms'c'}, z^b_{msc}) - d(z^b_{ms'c'}, z^b_{m's'c'}) + \delta]_+, \quad (1)$$

where $d(\cdot)$ is a distance metric, δ is a margin between X^b_{msc} and $X^b_{m's'c'}$ pair, and $[\cdot]_+$ is a hinge function [51]. The triplet loss makes the distance between a reference and a positive sample close, while it enforces the distance between a reference and a negative sample to be far. However, this loss does not contain information on how similar the samples are.

To overcome this limitation of the triplet loss, we propose a loss term by utilizing a motion variation score between samples in the same action category. The proposed motion

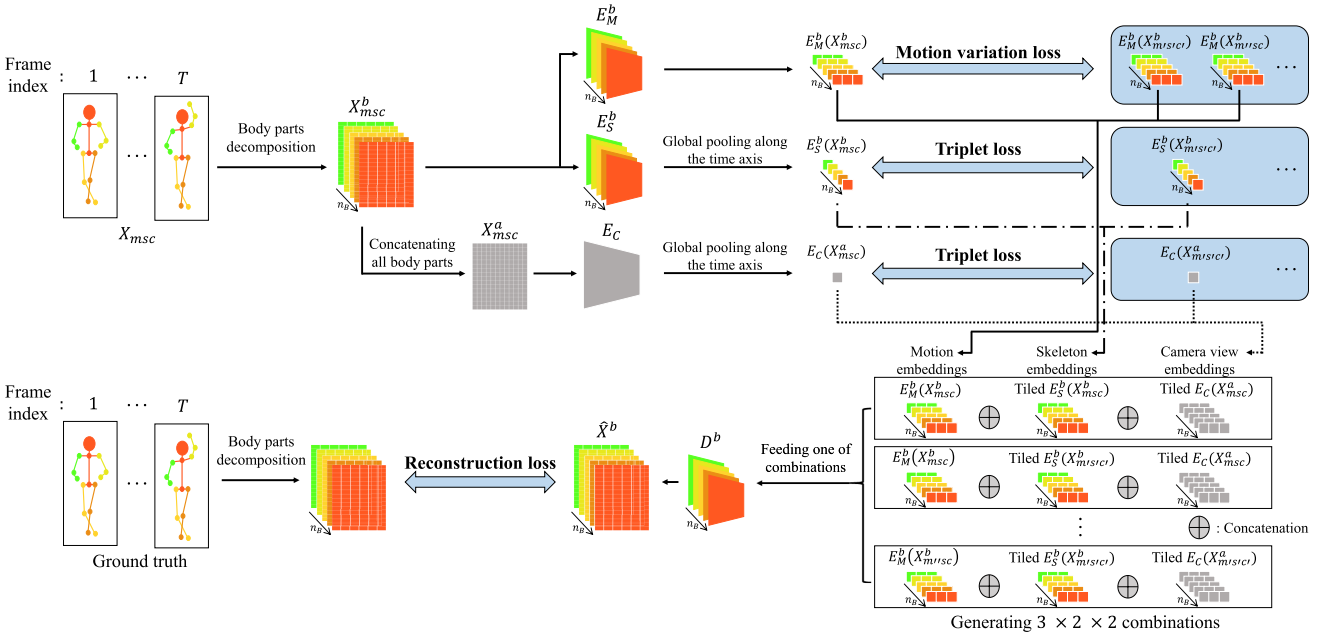


FIGURE 4. Visualization of the proposed model. Each body part is drawn in a different color.

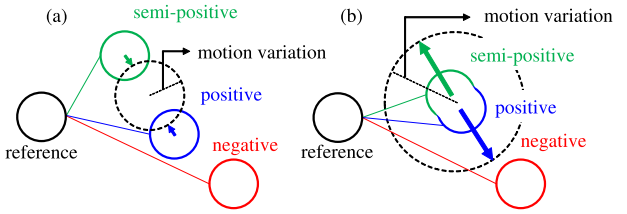


FIGURE 5. Visualization of the motion variation loss. (a) shows a situation where loss brings positive and semi-positive samples closer when they are mapped far; (b) indicates that the loss drives them far when they are mapped close.

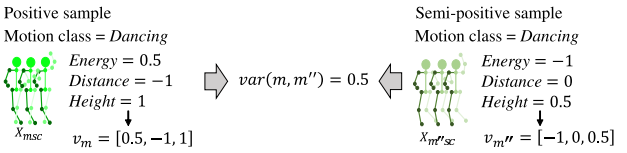


FIGURE 6. Visualization of the motion variation for positive and semi-positive samples of the SARA dataset. The motion variation is computed from two samples that belong to the same motion class but have different characteristics (e.g., *Energy*).

variation loss projects positive and semi-positive samples at a certain distance defined by the motion variation, as illustrated in Fig. 5. Assuming that there are variables that can control the movement of the skeleton, such as *Energy*, *Distance*, and *Height* in Fig. 6, we let v_m be the characteristic vector that has each element corresponding to one of these variables for motion m. Since m and m'' belong to the same motion class, v_m and $v_{m''}$ have the same n_{v_m} number of variables. Then, the motion variation $var(m, m'')$ between m and m'' is defined as:

$$var(m, m'') = \frac{\|v_m - v_{m''}\|_1}{2 \times n_{v_m}}. \tag{2}$$

The motion variation loss $\mathcal{L}^b_{var}$ is defined by using the motion variation as:

$$\begin{aligned}\mathcal{L}^b_{var}(X^b_{msc}, X^b_{m's'c'}, X^b_{m''sc}) \\ = \mathcal{L}^b_M(X^b_{msc}, X^b_{m's'c'}) + \mathcal{L}^b_M(X^b_{m''sc}, X^b_{m's'c'}) \\ + \alpha\{d(z^b_{msc}, z^b_{m''sc}) - \beta \cdot var(m, m'')\}^2,\end{aligned} \tag{3}$$

where $d(\cdot)$ is a distance metric, and hyper-parameters α and β are respectively set 1 and 0.1 in our experiments. With this loss term, we expect the motion embedding vectors of positive and semi-positive samples to be dependent on the characteristic vector.

For the skeleton and camera view embeddings, triplet losses $\mathcal{L}^b_S$ and $\mathcal{L}_C$ can be obtained in the same manner as (1). These terms are then combined to complete the final similarity loss term:

$$\mathcal{L}_{sim} = \sum_{b \in B} \mathcal{L}^b_{var} + \sum_{b \in B} \mathcal{L}^b_S + \mathcal{L}_C. \tag{4}$$

The estimate $\hat{X}^b \in \mathbb{R}^{2 \times n_b \times T}$, which is the output of the D^b, can be obtained by providing the concatenation of motion, skeleton, and camera view embedding vectors to D^b.

Algorithm 1: Measuring Motion Similarity

Input : motion sequences X_1, X_2,
motion encoders
$E_M = \{E^1_M, \cdots, E^b_M, \cdots, E^{n_B}_M\}$,
video sampling window size w,
video sampling stride r
Output: similarity sim

divide X_1, X_2 into each body part $\{X^1_1, \cdots, X^b_1, \cdots, X^{n_B}_1\}$, $\{X^1_2, \cdots, X^b_2, \cdots, X^{n_B}_2\}$
for $b = 1$ **to** n_B **do**
  extract patches from X^b_1, X^b_2 using sliding window with w, r
  obtain embeddings F^b_1, F^b_2 from the extracted patches by using E_M
  $path \leftarrow DTW(F^b_1, F^b_2)$
  $sim^b \leftarrow$ average cosine similarity between the embedding pairs in $path$
$sim \leftarrow$ the average of $\{sim^b | b \in B\}$

While the considered BPE model is able to accommodate 12 combinations specified by the different attributes of M, S, and C as inputs, only three inputs, X^b_{msc}, $X^b_{m's'c'}$, and $X^b_{m''sc}$ are utilized to calculate reconstruction error for computational efficiency. Specifically, motion embeddings, $E^b_M(X^b_{msc})$, $E^b_M(X^b_{m's'c'})$, and $E^b_M(X^b_{m''sc})$, skeleton embeddings, $E^b_S(X^b_{msc})$ and $E^b_S(X^b_{m's'c'})$, and camera view embeddings, $E_C(X^a_{msc})$ and $E_C(X^a_{m's'c'})$ are concatenated into 12 different ways to build the inputs of D^b. Before the concatenation, the camera view embedding is copied by the number of body parts. Then, the skeleton embeddings and all the copied camera view embeddings are tiled along the time axis. The reconstruction error can then be calculated by comparing the output $\hat{X}^b$ of the decoder D^b with the ground truth. The reconstruction error for each body part is defined as follows:

$$\mathcal{L}^b_{rec} = \frac{1}{12} \sum_{i \in M} \sum_{j \in S} \sum_{k \in C} (\hat{X}^b_{ijk} - X^b_{ijk})^2. \tag{5}$$

This reconstruction error term helps disentangle motion, skeleton, and camera view embedding vectors.

Finally, foot velocity loss $\mathcal{L}_f$ used in [19] is applied to prevent a foot skating phenomenon that causes a significant error in hands and feet. The final loss is a weighted sum of the individual loss terms:

$$\mathcal{L} = \lambda_1 \sum_{b \in B} \mathcal{L}^b_{rec} + \lambda_2 \mathcal{L}_{sim} + \lambda_3 \mathcal{L}_f, \tag{6}$$

where the weights λ_1, λ_2, and λ_3 are respectively set 1, 1, and 0.5 in our experiments.

B. MEASURING MOTION SIMILARITY

The measurement of similarity between motions is described in Algorithm 1. We use the outputs of the motion encoders only so that the model can generate predictions robust to differences in view-points or skeletons (e.g., different heights).

Let two sequences $X_1 \in \mathbb{R}^{2\times J\times T_1}$ and $X_2 \in \mathbb{R}^{2\times J\times T_2}$, which are targets for measuring motion similarity, have time lengths of T_1 and T_2, respectively. Also let $X_1^b \in \mathbb{R}^{2\times n_b\times T_1}$ and $X_2^b \in \mathbb{R}^{2\times n_b\times T_2}$ respectively be the sequences corresponding to body part b of X_1 and X_2. Sliding window approach with window size w and stride r is applied to split X_1^b and X_2^b into patches, which are then put into the motion encoder E_M^b. Let F_1^b and F_2^b respectively denote the sets of motion embeddings of X_1^b and X_2^b patches. They are then fed as inputs to the DTW algorithm to determine the best alignment between the sequences. Subsequently, the similarity of each body part can be obtained through average cosine similarity between matching time frames on the DTW path. Based on each body part's similarity, the final similarity of two sequences X_1 and X_2 can be calculated by averaging over body parts and temporal timestamps.

IV. DATASETS

Training the model to produce a motion embedding representation is performed with a synthetically created 3D motion dataset. Additionally, to demonstrate the generalization capabilities of the proposed model in evaluating motion similarity of the real-world data, we have manually annotated the NTU RGB+D 120 dataset. The latter is only used for performance evaluation.

A. SYNTHETIC MOTION DATASET: SARA DATASET

We have constructed a 3D motion dataset, named Synthetic Actors and Real Actions (SARA), for training a model to produce motion embeddings suitable for reasoning about motion similarity. For this, Mixamo [20] has been utilized.

Motion sequence data was generated by combining 18 different actors (i.e., action performing characters). The characters were rendered in a skeleton shape with Adobe Fuse software. We select four action categories (*Combat*, *Adventure*, *Sport*, and *Dance*) comprising a number of motion variations, where each action has a frame length of 32 or more. There are 4,428 base motions (e.g., dancing, jumping) in the SARA dataset. With these motions, the intra-class variations were generated. Mixamo allows the users to control various characteristics of each motion (e.g., *Energy*) that can be adjusted to create different motion's characteristic. The value of the characteristics variables is within the range of $[-1, 1]$, and in the SARA dataset, it is set to one of $\{-1, -0.5, 0, 0.5, 1\}$. This parameter is configured differently according to a motion. Each sequence frame provides 3D coordinates of 17 joints from all body parts, and we have generated samples through 2D projection. The statistics for the dataset are summarized in Table 1.

B. NTU RGB+D 120 SIMILARITY ANNOTATIONS

We have collected motion similarity annotations for NTU RGB+D 120 dataset to evaluate motion similarity in the real world. The NTU RGB+D 120 dataset is an action recognition dataset consisting of 114,480 videos covering 120 different actions of 106 people. While original videos of the dataset were used to obtain ground truth motion similarity from AMT, only the 2D skeleton sequences are utilized in our model to estimate the motion similarity.

TABLE 1. SARA dataset overview.

Action category	The number of characters	The number of base motions	The number of variations
Combat	18	3,000	76,512
Adventure		264	3,390
Sport		306	4,485
Dance		858	18,756
Total	18	4,428	103,143

Only a portion of the entire dataset has been utilized since the actions with little movements such as reading, writing, and phone call also exist in the original NTU RGB+D 120 dataset. After filtering out these actions, 21 actions with large and well-defined movements were selected based on visual inspection. Then, two videos of 39 people for each action were sampled. The total number of sampled video clips were 1,638 (21 actions $\times$ 39 people $\times$ 2 videos).

We have obtained the motion similarity scores from humans of AMT [32] by using the sampled videos. The motion similarity was scored on a 4-point scale ranging from 1 (utterly different motions) to 4 (same movements) for each pair of video clips. The similarity score for a pair is an average of scores collected from at least ten workers of AMT. Among all the possible candidates, the annotations for 20,093 randomly sampled video pairs were collected. We use all the annotations to evaluate the models, not to train. These annotations are released on our project page. More detailed information, including instructions and annotation guidelines provided to the workers, can be found in the supplementary material. Some of the video pairs and distribution of the annotated similarity scores are shown in Fig. 7.

There are some imprecise skeleton data in NTU RGB+D 120. To cope with this problem and to generate new 2D joint annotations, we have used our reproduction of Multi-PoseNet [52] with the average precision of 0.709 for large objects in COCO 2017 valid set to generate new 2D joint annotations.

V. EXPERIMENTS

In this section, we first present implementation details for our model, then introduce the correlation measurements between the collected annotations for NTU RGB+D 120 pairs and the similarities produced by several models, including ours and baselines'. Next, we visualize the motion latent space of our model. Finally, we explain how our framework can be applied to real-world tasks. For all the experiments in this section, only the SARA dataset is used for training, and the NTU RGB+D similarity annotations are used for evaluation.

A. IMPLEMENTATION DETAILS

1) PREPROCESSING

First, all motion sequences are divided into segments of 32 frames. Then we split the SARA dataset into training and validation sets composed of different base motions of

FIGURE 7. The examples of the AMT annotations pair from NTU RGB+D 120 [22] dataset. (a), (b), (c), and (d) are an example of scores 4, 3, 2, and 1 respectively; (e) is the histogram of the total collected scores.

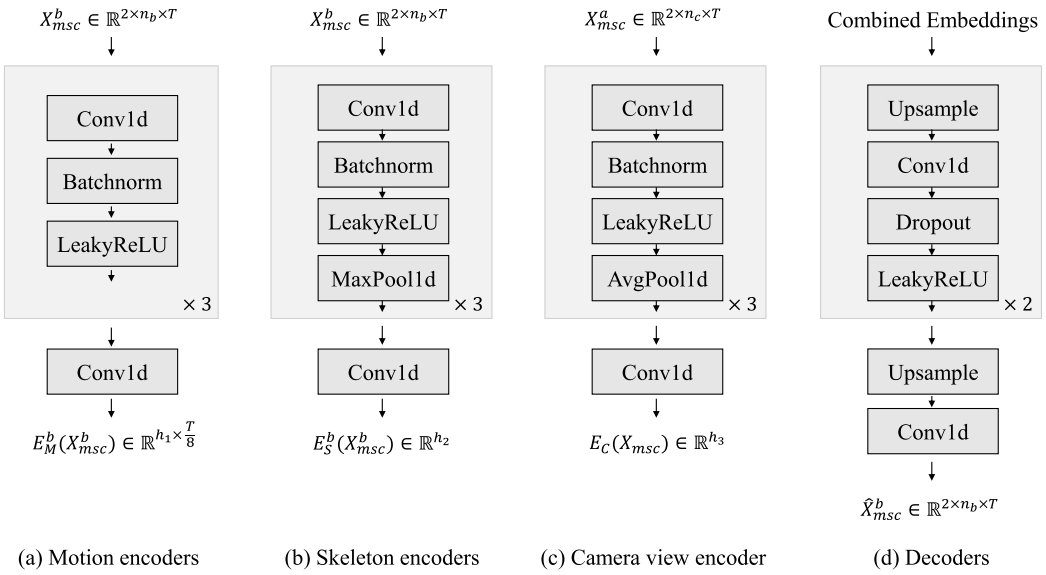


FIGURE 8. Network structure of encoders and decoders. Except for the camera view encoder, encoders and decoders are as many as the number of body parts.

non-overlapping characters. As a result, 455,028 32-frames motions from 12 characters and 64,218 32-frames motions from 6 characters are constructed for training and validation, respectively.

In a real-world environment, the size of a person's projection varies depending on the distance from a camera. To handle this problem, the skeleton size of the sequence is reduced or increased with the scale factor, which is randomly sampled between 0.5 and 1.5. After the scale adjustment, the reference joint for each body part is selected, and the coordinates of all joints are changed from absolute to relative coordinates. Reference joints for each body part are: shoulders for arms, hips for legs, and the middle hip for a torso, respectively. Finally, the coordinates are normalized using the z-normalization to produce the final input.

2) NETWORK STRUCTURE

The encoders and decoders of our BPE model depicted in Fig. 8 are implemented as convolutional layers with a batch normalization [53] layer and a leaky rectified linear unit (Leaky ReLU) [54] activation function in between each layer. The motion encoder for each body part takes a 2D sequence for the corresponding body part as input. Let $X^b_{msc} \in \mathbb{R}^{2\times n_b\times T}$ be a sequence fed into the encoder for body part b. Each motion encoder generates the embedding, which is denoted as $E^b_M(X^b_{msc}) \in \mathbb{R}^{h_1\times \frac{T}{8}}$, where $h_1 = 128$ for the torso motion encoder and $h_1 = 64$ for the other encoders. The torso embedding is set to a higher dimension since the number of joints is greater than other body parts. In the case of the skeleton encoders, the input is the same as the motion encoders. The difference is that it generates an embedding that compresses the temporal information using global max pooling. This embedding, denoted as $E^b_S(X^b_{msc}) \in \mathbb{R}^{h_2}$, has a dimension $h_2 = 32$ for torso and $h_2 = 16$ for the others. The camera view encoder uses the concatenation of the body parts as input. Unlike the skeleton encoder, we use average pooling to make the embedding with dimension $h_3 = 64$. The camera view embedding is copied by the number of body parts. Then, the generated skeleton and camera view embeddings are tiled along the time axis to match the size of the motion embedding, $\frac{T}{8}$, and subsequently concatenated. The decoder then yields an estimate $\hat{X}^b_{msc}$ for each body part

TABLE 2. The rank correlations with AMT scores.

Joints annotations	NTU RGB+D 120 [22]		Pose estimated joints	
Method	Original pair	Body flip	Original pair	Body flip
Kim and Kim [9]	0.1692	0.1601		
Joint distance	0.2222	0.1721	0.2634	0.1948
Coskun *et al.* [4]	0.2252	0.2335	0.2845	0.2996
Aberman *et al.* [19]	0.3207	0.3341	0.3545	0.3911
BPE (ours)	0.4345	0.4609	**0.5509**	**0.5970**

TABLE 3. The ablation study on the proposed loss function.

Method	With variation	With recons.	Original pair	Body flip
BPE (ours)	X	X	0.5264	0.5662
	X	O	0.5280	0.5740
	O	X	0.5363	0.5758
	O	O	**0.5509**	**0.5970**

by utilizing the concatenated embeddings. The size of the resulting output is the same as the input size of the encoders.

3) OPTIMIZATION

The model was trained using Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. $L2$ regularization with weight decay of 0.01 was also used to prevent overfitting. The initial learning rate was 10^{-3}, and we applied an exponential decay with a rate of 0.98 every 1/3 epoch. Using a single GPU (NVIDIA Tesla V100) and Intel Xeon 5120 @ 2.20GHz, it took less than 20 minutes for the model to train 1 epoch with 12 workers and a batch size of 2,048.

4) MODEL SELECTION

As mentioned above, we trained the model only with the SARA dataset. The model parameters to be employed for evaluation were selected based on the epoch with the lowest total loss for the SARA validation set.

B. NTU RGB+D 120 SIMILARITY ANNOTATIONS

1) COMPARISONS WITH THE OTHER BASELINES

We calculated the correlations between the model's predictions and the annotated similarity scores to determine how comparable the model is to human perception. Spearman's rank correlation was employed as an evaluation metric, and 20,093 pairs were used to obtain the correlations. For the models relying on 2D coordinates, the symmetrical nature of a human body was utilized. In detail, each model predicted two different similarity scores for each motion pair: one from the original motion sequences and the other by using horizontally flipped ones from the two sequences, and then the larger value was selected as the final similarity prediction. It is referred to as *Body flip* in Table 2 and Table 3.

Four approaches were considered as baselines. The first one was a heuristic algorithm that calculated the Euclidean distance between the joints of the matching frames. DTW was used to align the frames of two motion sequences. As the second baseline, we incorporated the algorithm of [9], where the authors proposed the similarity of 3D motion sequences between teacher and learner in a dance teaching situation. We used the 3D joint coordinates of the NTU RGB+D 120 as inputs for this method.

TABLE 4. Motion similarity by body part for the sample pairs in Fig. 9. Body parts with relatively lower similarity scores are marked bold.

	Right Arm	Left Arm	Right Leg	Left Leg	Torso
(a)	**-0.0153**	0.6982	0.9297	0.9205	0.9429
(b)	**0.1740**	0.7338	0.8843	0.6937	0.8773
(c)	**0.1841**	**0.0648**	0.9423	0.9321	0.9160
(d)	0.8196	0.7210	**0.3999**	**0.5499**	**0.2366**
(e)	**0.2147**	**-0.2170**	**0.1040**	**0.1517**	**0.1057**

As another baseline, we considered the approach of Coskun *et al.* [4], which carried out tasks of action recognition and retrieval, since it learns the similarity between the motions through metric learning. We re-implemented the model of [4] and trained it with the SARA dataset for a fair comparison. Similar to Algorithm 1, DTW was employed to align the motion embedding patches. Finally, we have trained the model of Aberman *et al.* [19] on the SARA dataset and used its motion embeddings for similarity measurement. Since [19] generated one motion embedding for the entire body's motion sequence, Algorithm 1 was modified to calculate the similarity for individual embedding vector.

Overall, the highest correlation results were achieved by the proposed BPE model, as shown in Table 2. Our method significantly improved the correlation results between the similarity estimation and the human perception. There are two main reasons for this. One is that our method can estimate the similarity score for each body part. When people compare two motions, they tend to think that the whole body performs different motions even if only one body part moves differently. Furthermore, our loss term allows the model to catch subtle intra-class variations and enables similarity estimation to be closer to human perception. The ablation study related to the loss term will be discussed in the following subsections.

Interestingly, in all the cases based on the proposed BPE model, the similarity correlation results produced with the *Body flip* had the best performance. This implies that horizontally flipped motions are considered as the same motions in the human perspective. For example, people may not care whether a human throws a ball with his or her right hand and tend to focus on the fact that a ball is thrown in determining whether the motions are similar.

Finally, we noted that using motion sequences corrected by MultiPoseNet results in a higher correlation score for every method. We believe that refining imprecisely annotated poses had an impact on this.

2) ABLATION STUDY

To verify the effectiveness of the proposed loss term, we have carried out ablation experiments. Specifically, we respectively removed the reconstruction and motion variation losses by excluding their contributions from the total loss. The results are outlined in Table 3.

The results show that the correlation scores increased when motion variation loss was applied. Unlike the triplet loss, motion variation loss forces the model to ensure that motion embeddings are separated even for slightly different motions

FIGURE 9. The pairs (from NTU RGB+D 120 [22]) for body part similarity in Table 4.

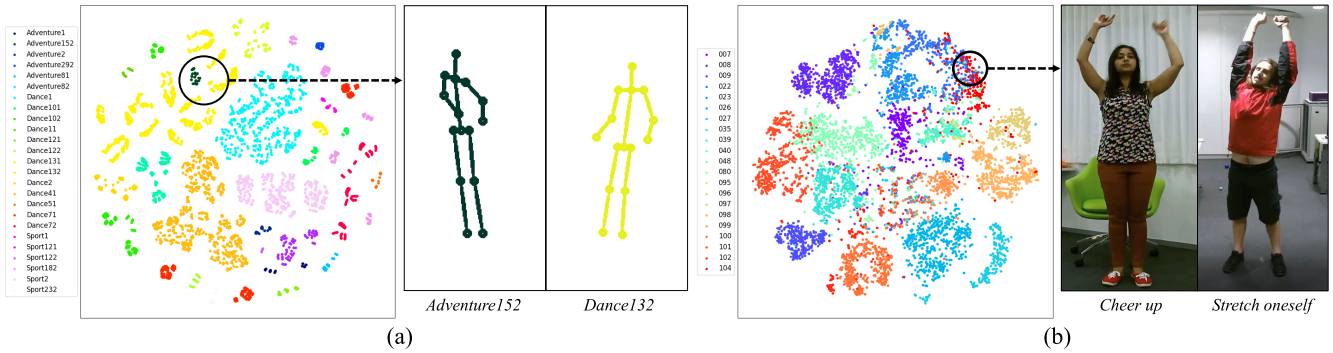


FIGURE 10. Visualization of motion latent vectors. The motion classes of the SARA validation set are clustered by colors in the left part of (a). The dark green (*Adventure152*) and light yellow (*Dance132*), circled in black, correspond to the similar motions that were performed while standing with the elbows bent and leaning back (shown in the right part of (a)). The visualization of 21 sampled actions of NTU RGB+D 120 [22] is made in the left part of (b). The blue (*cheer up*) and red (*stretch oneself*) positioned on the upper right, represent similar motions (shown in the right part of (b)).

from the same class. We argue that this property helps a model to generate similarity predictions close to human perception.

When we omitted the reconstruction loss, the correlation scores decreased. We claim that the reconstruction loss of our model enforces the embedding to contain essential information of the motions, and when it is applied with a cross-reconstruction scheme, it can generate the embedding of the motion attribute independent of the skeleton or camera view.

The BPE model without the motion variation loss (the second row of Table 3) has shown better performances than Aberman *et al.* (the third row of Table 2), suggesting the effectiveness of the proposed body part decomposition approach. The motion embedding for each body part appears to make it possible for the model to capture the detailed motion information.

3) MOTION SIMILARITY COMPARISON BY BODY PART

Our model computed the motion similarity for each body part for a given pair of motions. Representative results from NTU RGB+D 120 are given in Table 4 with the visual references in Fig. 9. Fig. 9 (a) represents a case where both people raise their left hand. Our model predicted high similarity results in most body parts except the right hand for which their positions were different. Next, Fig. 9 (b) shows a motion where one person raises both hands while the other raises a left hand. The model predicted a lower similarity score for the right arm while having a high score for the remaining parts. Fig. 9 (c) represents motion sequences with the same waving motion performed with a different hand. It was found that the motion similarities for both arms were lower than for the other body parts. In Fig. 9 (d), the left person sits on the chair, and the right one performs squats. The model predicted the similarities of legs and torso lower than those of arms since the angles of the knees and hips were different. Finally, an example of a comparison between a person standing with the raised arms and a person sitting is displayed in Fig. 9 (e). The similarity scores in all parts were relatively low as all body parts' motions do not match.

C. VISUALIZATION OF MOTION LATENT CLUSTERS

The motion latent spaces for the SARA validation set and NTU RGD+D 120 are shown in Fig. 10, visualized using t-SNE [55]. Fig. 10 (a) shows that despite the differences in the characters and camera views in the SARA dataset, the sequences with the same motion attributes are clustered together. It supports the claim that the similarity can be measured by considering only the motion, independent of the humans or camera views. Furthermore, motions that were closely mapped corresponded to similar movements of body parts as displayed on the right side of Fig. 10 (a), even though they belonged to different classes. The plot on the left of

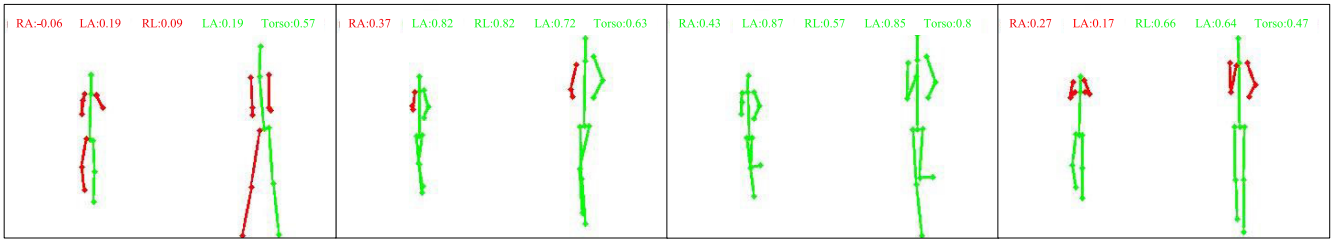


FIGURE 11. Illustration of comparing two dance sequences by each body part. Threshold of 0.4 is chosen to separate similar (green) and different (red) motions.

Fig. 10 (b) shows the motion latent space for the 21 sampled actions of NTU RGB+D 120. Overall, the samples with the same action class formed clusters. In some cases, however, different motions were mapped closely, similar to the aspect shown in the SARA dataset's case. The photos on the right of Fig. 10 (b) are examples of such cases in which they are positioned closely due to their motion classes' similarities from the human perspective.

D. APPLICATION

In this subsection, we provide a guideline on how to utilize the proposed motion similarity in the real-world. Evaluating an exercise (e.g., dance, yoga, and figure skating) performance is a natural application of the proposed model.

Let's assume that we have a dancer with a goal of repeating ground truth movements. Our goal is to assess the performance periodically as the dance progresses. For this purpose, we compared two temporally aligned videos of people trying to perform the same dance. To extract the joints' location, we used the method of [52] while any human pose estimation algorithm is suitable. Algorithm 1 with window size $w = 32$ and stride $r = 32$ was used to obtain the motion similarities between two sequences. The parameters were set to provide feedback approximately every second. However, they could be defined arbitrarily based on the application or user preference. Fig. 11 shows an example of interactive motion performance feedback. Several full video sequences are available in the supplementary materials.

To compare two video clips (3.5 minutes long, 24fps) the proposed method took about 7.8s (approx. 670fps) on CPU (Intel Xeon 5120 @ 2.20GHz) without model or code optimization. This included joints data preprocessing, network inference, and motion similarity calculation while excluding the pose estimation extraction.

Note that, in its current form, the proposed method neither provides feedback on the exact incorrect human joint location nor how to correct the location to make the action more similar. Yet, our method provides a similarity score on the sequences without requiring evaluation of the motion similarity based on the manually defined rules, for instance, normalized distances between joints of an actor and the ground truth.

VI. CONCLUSION

In this paper, we proposed a method of measuring similarity for two motion sequences. To compute similarity, we generated motion embedding vectors for each body part, and the motion variation loss term was introduced to distinguish similar motions. Additionally, a synthetic dataset to train the model was constructed. For the evaluation purposes, we have collected real-world annotations of the NTU RGB+D 120 dataset. The evaluation indicated that our method achieved the best performances compared to the other baseline models considered.

Since our approach depends on the motion sequence, the similarity model performs best when precise pose estimation is available. However, the pose estimation may not be satisfactory in challenging situations (e.g., occlusions and crowded scenes), and it is our future work to accurately measure the similarity even in those challenging situations. Extending the model to learn temporal alignment is also an important future work. We expect the extended model to produce better similarity predictions by using both aligned and non-aligned action datasets, and data-driven sequence alignment. Finally, evaluating the performance of existing tasks such as action recognition or person re-identification by applying the motion similarity is also viable. In fact, the aforementioned tasks already take advantage of the concept of motion similarity, and it would be interesting to see how the proposed method can contribute to those tasks.

ACKNOWLEDGMENT

Portions of the research in this paper used the NTU RGB+D 120 Action Recognition Dataset made available by the ROSE Lab at the Nanyang Technological University, Singapore. Jonghyuk Park also thanks Sungwook Jeon of the Seoul National University, for helpful discussion. The Institute of Engineering Research at Seoul National University provided research facilities for this work *(Jonghyuk Park, Sukhyun Cho, and Dongwoo Kim contributed equally to this work.)*

REFERENCES

[1] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on context-dependent graph kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2609–2616.
[2] A. Ciptadi, M. S. Goodwin, and J. M. Rehg, "Movement pattern histogram for action recognition and retrieval," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 695–710.
[3] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal vector of locally max pooled features for action recognition in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3097–3106.
[4] H. Coskun, D. J. Tan, S. Conjeti, N. Navab, and F. Tombari, "Human motion analysis with deep metric learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 667–683.

[5] Z. Gao, L. Guo, T. Ren, A.-A. Liu, Z.-Y. Cheng, and S. Chen, "Pairwise two-stream ConvNets for cross-domain action recognition with small data," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 9, 2020, doi:10.1109/TNNLS.2020.3041018.

[6] Z. Gao, L. Guo, W. Guan, A.-A. Liu, T. Ren, and S. Chen, "A pairwise attentive adversarial spatiotemporal network for cross-domain few-shot action recognition-R2," *IEEE Trans. Image Process.*, vol. 30, pp. 767–782, Nov. 2021.

[7] F. Chin-Shyurng, S.-E. Lee, and M.-L. Wu, "Real-time musical conducting gesture recognition based on a dynamic time warping classifier using a single-depth camera," *Appl. Sci.*, vol. 9, no. 3, p. 528, Feb. 2019.

[8] A. Elaoud, W. Barhoumi, E. Zagrouba, and B. Agrebi, "Skeleton-based comparison of throwing motion for handball players," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 1, pp. 419–431, Jan. 2020.

[9] Y. Kim and D. Kim, "Real-time dance evaluation by markerless human pose estimation," *Multimedia Tools Appl.*, vol. 77, no. 23, pp. 31199–31220, Dec. 2018.

[10] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation (SCA)*, Aug. 2011, pp. 147–156.

[11] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 243–255, Feb. 2015.

[12] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2909–2917.

[13] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, "Learning regularity in skeleton trajectories for anomaly detection in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11996–12004.

[14] X. Zhang, S. Yang, X. Zhang, W. Zhang, and J. Zhang, "Anomaly detection and localization in crowded scenes by motion-field shape description and similarity-based statistical learning," 2018, *arXiv:1805.10620*. [Online]. Available: http://arxiv.org/abs/1805.10620

[15] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee, "Part-aligned bilinear representations for person re-identification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 402–419.

[16] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6479–6488.

[17] L. Wu, Y. Wang, J. Gao, and X. Li, "Where-and-when to look: Deep siamese attention networks for video-based person re-identification," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1412–1424, Jun. 2019.

[18] W. Zhang, Y. Li, W. Lu, X. Xu, Z. Liu, and X. Ji, "Learning intra-video difference for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 3028–3036, Oct. 2019.

[19] K. Aberman, R. Wu, D. Lischinski, B. Chen, and D. Cohen-Or, "Learning character-agnostic motion for motion retargeting in 2D," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019, Art. no. 75.

[20] Adobe Systems. (2020). *Mixamo*. Accessed: Sep. 7, 2020. [Online]. Available: https://www.mixamo.com

[21] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.

[22] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.

[23] F. Li, A. Zhu, Y. Xu, R. Cui, and G. Hua, "Multi-stream and enhanced spatial-temporal graph convolution network for skeleton-based action recognition," *IEEE Access*, vol. 8, pp. 97757–97770, 2020.

[24] Y. Han, S.-L. Chung, Q. Xiao, W. Y. Lin, and S.-F. Su, "Global spatio-temporal attention for action recognition based on 3D human skeleton data," *IEEE Access*, vol. 8, pp. 88604–88616, 2020.

[25] H. Yang, Y. Gu, J. Zhu, K. Hu, and X. Zhang, "PGCN-TCA: Pseudo graph convolutional network with temporal and channel-wise attention for skeleton-based action recognition," *IEEE Access*, vol. 8, pp. 10040–10047, 2020.

[26] Y. Wang, Z. Xu, L. Li, and J. Yao, "Robust multi-feature learning for skeleton-based action recognition," *IEEE Access*, vol. 7, pp. 148658–148671, 2019.

[27] W. Nie, W. Wang, and X. Huang, "SRNet: Structured relevance feature learning network from skeleton data for human action recognition," *IEEE Access*, vol. 7, pp. 132161–132172, 2019.

[28] Y. Fan, S. Weng, Y. Zhang, B. Shi, and Y. Zhang, "Context-aware cross-attention for skeleton-based human action recognition," *IEEE Access*, vol. 8, pp. 15280–15290, 2020.

[29] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 143–152.

[30] H. Yang, D. Yan, L. Zhang, D. Li, Y. Sun, S. You, and S. J. Maybank, "Feedback graph convolutional network for skeleton-based action recognition," 2020, *arXiv:2003.07564*. [Online]. Available: http://arxiv.org/abs/2003.07564

[31] K. Papadopoulos, E. Ghorbel, D. Aouada, and B. Ottersten, "Vertex feature encoding and hierarchical temporal modeling in a spatial-temporal graph convolutional network for action recognition," 2019, *arXiv:1912.09745*. [Online]. Available: http://arxiv.org/abs/1912.09745

[32] Amazon Mechanical Turk, Amazon. (2020). Accessed: Sep. 7, 2020. [Online]. Available: https://www.mturk.com

[33] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Pose search: Retrieving people using their pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1–8.

[34] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar, "Video retrieval by mimicking poses," in *Proc. 2nd ACM Int. Conf. Multimedia Retr. (ICMR)*, 2012, pp. 1–8.

[35] A. Aristidou, D. Cohen-Or, J. K. Hodgins, Y. Chrysanthou, and A. Shamir, "Deep motifs and motion signatures," *ACM Trans. Graph.*, vol. 37, no. 6, Dec. 2018, Art. no. 187.

[36] Y. Shen, L. Yang, E. S. L. Ho, and H. P. H. Shum, "Interaction-based human activity comparison," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 8, pp. 2620–2633, Aug. 2020.

[37] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[38] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose MoTion representation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7024–7033.

[39] X. Guo and J. Choi, "Human motion prediction via learning local structure representations and temporal dependencies," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2018, pp. 7024–7033.

[40] A.-A. Liu, Y.-T. Su, P.-P. Jia, Z. Gao, T. Hao, and Z.-X. Yang, "Multiple/single-view human action recognition via part-induced multitask structural learning," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1194–1208, Jun. 2015.

[41] Y.-L. Li, L. Xu, X. Liu, X. Huang, Y. Xu, M. Chen, Z. Ma, S. Wang, H.-S. Fang, and C. Lu, "HAKE: Human activity knowledge engine," 2019, *arXiv:1904.06539*. [Online]. Available: http://arxiv.org/abs/1904.06539

[42] N. Jammalamadaka, A. Zisserman, and C. V. Jawahar, "Human pose search using deep networks," *Image Vis. Comput.*, vol. 59, pp. 31–43, Mar. 2017.

[43] G. Mori, C. Pantofaru, N. Kothari, T. Leung, G. Toderici, A. Toshev, and W. Yang, "Pose embeddings: A deep architecture for learning to match human poses," 2015, *arXiv:1507.00302*. [Online]. Available: http://arxiv.org/abs/1507.00302

[44] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[45] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3109–3118.

[46] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: http://arxiv.org/abs/1703.07737

[47] S. Kim, M. Seo, I. Laptev, M. Cho, and S. Kwak, "Deep metric learning beyond binary supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2288–2297.

[48] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 403–412.

[49] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5620–5629.

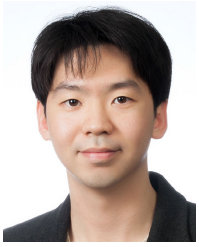
[50] R. Bellman and R. Kalaba, "On adaptive control processes," *IRE Trans. Autom. Control*, vol. 4, no. 2, pp. 1–9, Nov. 1959.
[51] C. Gentile and M. K. Warmuth, "Linear hinge loss and average margin," in *Proc. Adv. Neural Inf. Process. Syst.*, Jul. 1999, pp. 1–7.
[52] M. Kocabas, S. Karagoz, and E. Akbas, "MultiPoseNet: Fast multi-person pose estimation using pose residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 417–433.
[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 448–456.
[54] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2013, pp. 1–6.
[55] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

JONGHYUK PARK received the B.S. degree in industrial engineering from Seoul National University (SNU), South Korea, in 2015, where he is currently pursuing the Ph.D. degree with the Laboratory of Information Management, Department of Industrial Engineering. His current research interests include computer vision and machine learning applications.

SUKHYUN CHO received the B.S. degree in industrial and systems engineering and biological science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2016. He is currently pursuing the Ph.D. degree with the Laboratory of Information Management, Department of Industrial Engineering, Seoul National University (SNU). His current research interests include natural language understanding, computer vision, and machine learning applications.

DONGWOO KIM received the B.S. and M.S. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2011 and 2015, respectively. In addition to five years of experience in engineering with Kakao, he is currently a Research Engineer with Kakao Brain, where he is working on building deep learning model and implementing on applications.

OLEKSANDR BAILO received the B.S. degree in electrical engineering and business and technology management and the M.S. degree in robotics and computer vision from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2015 and 2017, respectively. He is currently a Research Engineer with Kakao Brain, where he is working on human pose estimation and motion similarity.



HEEWOONG PARK received the B.S. degree in statistics from Seoul National University (SNU), South Korea, in 2013, where he is currently pursuing the Ph.D. degree with the Laboratory of Information Management, Department of Industrial Engineering. His current research interests include natural language understanding, user profiling, and machine learning applications.

SANGHOON HONG received the B.S. degree in electrical engineering and the M.S. degree in bio and brain engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2010 and 2011, respectively. He is currently working with Kakao Brain, South Korea, as a Research Engineer. His current research interests include computer vision, deep neural networks, and their applications.

JONGHUN PARK received the Ph.D. degree in industrial and systems engineering with a minor in computer science from the Georgia Institute of Technology, Atlanta, in 2000. He was with the School of Information Sciences and Technology, The Pennsylvania State University, University Park, and the Department of Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, both as an Assistant Professor. He is currently a Professor with the Department of Industrial Engineering, Seoul National University (SNU), South Korea. His research interests include generative artificial intelligence and deep learning applications.