IEICE TRANS. INF. & SYST., VOL.E101–D, NO.6 JUNE 2018

PAPER

Data Augmented Dynamic Time Warping for Skeletal Action Classification

Ju Yong CHANG[†] ***and*** **Yong Seok HEO**[††a)]**,** ***Members***

SUMMARY We present a new action classification method for skeletal sequence data. The proposed method is based on simple nonparametric feature matching without a learning process. We first augment the training dataset to implicitly construct an exponentially increasing number of training sequences, which can be used to improve the generalization power of the proposed action classifier. These augmented training sequences are matched to the test sequence with the relaxed dynamic time warping (DTW) technique. Our relaxed formulation allows the proposed method to work faster and with higher efficiency than the conventional DTW-based method using a non-augmented dataset. Experimental results show that the proposed approach produces effective action classification results for various scales of real datasets.

key words: *action classification, dynamic time warping, data augmentation*

1. Introduction

Human action recognition, with its various application areas, such as human-computer interaction, visual surveillance, and intelligent robots, has been actively researched over the past several decades in various fields, including computer vision, pattern recognition, and machine learning. Recent surveys have been conducted on action recognition from RGB video [1] and depth maps [2]. With the advent of cost-effective depth cameras [3] and efficient 3D human pose estimation methods [4], skeleton-based action recognition approaches have recently gained increasing attention [5]–[12]. The human skeleton information, which has been frequently used to represent the human body, is known to provide effective high-level features for action recognition [13].

Among the different types of methods, the dynamic time warping (DTW)-based ones [7], [10] provide simple and intuitive solutions for action recognition. DTW, which has been frequently used in speech recognition [14], can measure the similarity between two temporal sequences. Considering that nonlinear alignment between two sequences can be achieved, DTW-based methods are robust to intra-class variations caused by varying the relative speeds at which the actions are performed. DTW can be used as a similarity function for the nearest neighbor (NN) search [10] and as a pre-processing step for learning-based classification [7].

The performance of the DTW-based methods generally depends on the size and diversity of training data. However, building a large-scale dataset with annotation is usually costly and sometimes impossible. Therefore, in this paper, we propose a novel *data augmented dynamic time warping (DA-DTW)* method for skeletal action classification. To increase the number of training samples, we adopt the idea of *data augmentation*, which has been commonly used to build better models by artificially enlarging a dataset [15]. Using the naive Bayes assumption, from the training dataset, we implicitly construct an exponentially increasing number of action sequences, which are matched to the test sequence using the DTW algorithm. Despite an enormous increase in the number of training sequences, the complexity of our DTW-based NN search with an augmented dataset can be significantly reduced to less than that of a conventional method with the original non-augmented dataset. The proposed method was tested on various scales of real datasets. Furthermore, it produces significantly improved results compared with conventional DTW-based approaches.

An overview of the proposed approach is illustrated in Fig. 1. The training set $\mathcal{Y}_c$ for action class c consists of N training sequences of length M, as shown in Fig. 1 (a). From them, we construct the augmented training set in Fig. 1 (b) by applying the data augmentation process, where a new sequence is created by combining the training sequences. More specifically, each frame-level feature of an augmented sequence is selected from among the set of all training features in that frame. Please see how an augmented sequence $\tilde{\mathbf{Y}}^{(i)} = (\tilde{\mathbf{y}}_1^{(i)} = \mathbf{y}_1^{(2)}, \tilde{\mathbf{y}}_2^{(i)} = \mathbf{y}_2^{(N)}, \ldots, \tilde{\mathbf{y}}_M^{(i)} = \mathbf{y}_M^{(1)})$ in Fig. 1 (b) is formed from the original training set in Fig. 1 (a). The input test sequence $\mathbf{X}$ in Fig. 1 (c) is then matched to all sequences in the augmented training set $\tilde{\mathcal{Y}}_c$ with the DTW algorithm, which produces the optimal output action class $\hat{c}$ as in Fig. 1 (d). Our relaxed DTW formulation can significantly reduce the complexity of this matching process.

Our contributions are as follows: First, we introduce data augmentation to increase the generalization capabilities of the conventional DTW-based classification approach. Second, an efficient DTW-based NN search method is presented based on *relaxation*, which can reduce the exponential time complexity to a logarithmic complexity. Third, our proposed method achieved a good performance in experiments with various real datasets.

Manuscript received August 31, 2017.
Manuscript revised January 20, 2018.
Manuscript publicized March 1, 2018.

[†]The author is with the Department of Electronics and Communications Engineering, Kwangwoon University, Korea.
[††]The author is with the Department of Electrical and Computer Engineering, Ajou University, Korea.
a) E-mail: ysheo@ajou.ac.kr

DOI: 10.1587/transinf.2017EDP7275

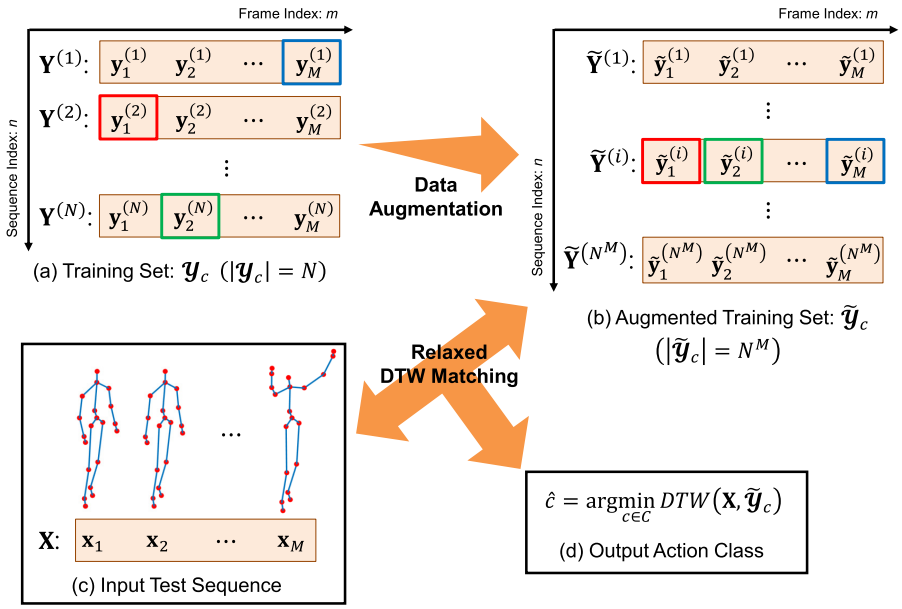


Fig. 1 An overview of the proposed method.

This paper is organized as follows. Section 2 introduces related works, Sect. 3 presents the proposed method, Sect. 4 outlines the experimental results, and Sect. 5 provides some concluding remarks.

2. Related Works

In this section, we review recent action recognition and data augmentation approaches that use skeletal data. Ellis and co-workers [5] addressed the latency issue for action recognition, and proposed a skeleton-based method in which a latency-aware learning framework is formulated to train a logistic regression-based classifier. It can automatically determine distinctive canonical poses from data in a similar fashion as multiple instance learning. Zhao and co-workers [12] proposed a new feature, namely, structured streaming skeletons, for online gesture recognition from skeletal sequence data to deal with the intra-class variations of gestures. Wu and Shao [9] proposed a deep neural network-based skeletal action recognition approach, in which the hidden Markov model is combined with a deep belief network to estimate better emission probabilities compared with a Gaussian mixture model. Wang and others [8] proposed an actionlet ensemble model for human action recognition with depth cameras. In their model, a subset of joints is introduced as an actionlet feature, and discriminative actionlets are discovered using a data mining approach. Vemulapalli and co-workers [7] proposed a new skeletal representation, in which human actions are modeled as curves in the Lie group. These are mapped to the Lie algebra and used for classification.

Data augmentation has been commonly used to reduce overfitting for parametric models with a large learning capacity, such as deep convolutional neural networks [15]. This generates additional training data by using label-preserving transformations, and the enlarged dataset is then used for model learning. Common transformations for image data include image translations, horizontal reflections, and altering the intensities of the RGB channels [15]. Rogez and Schmid [16] proposed a data augmentation method that can generate synthetic images with 3D human pose annotations by using 3D motion capture data and an existing real image dataset with 2D pose annotations. The resultant artificial images can be used to train a convolutional neural network for 3D human pose estimation, which generalizes well to real images and outperforms other state-of-the-art approaches. DeVries and Taylor [17] proposed a domain-agnostic approach to data augmentation, in which simple transformations of existing data such as noise addition, interpolation, and extrapolation are performed in a learned feature space, not in input space. The proposed method was applied to several datasets from different domains such as speech, images, and 3D skeletons. And it was shown that the augmented dataset generated through this can help in improving the performance of the supervised learning algorithm.

3. Proposed Method

A skeletal action can be represented as a sequence of frame-level features $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)$, where $\mathbf{z}_m$ denotes the skeletal feature vector at m-th frame. In the skeletal feature, we simply use the joint position feature, which is defined as the concatenation of 3D coordinates of all joints. For the invariance of our feature to the location of the person, we normalize the feature by subtracting the position vector of the particular reference joint (for example, the hip center or neck) from all joint vectors. Despite its simplicity, the joint position feature achieves a comparable action recognition performance with other sophisticated features [7]. To in-

crease the discriminability and achieve a more robust DTW alignment, the feature vectors from W frames near the m-th frame are concatenated into a new frame-level feature $\mathbf{x}_m$ as follows:

$$\mathbf{x}_m = \left[\mathbf{z}^T_{m-\lceil W/2\rceil+1}, \ldots, \mathbf{z}^T_{m+\lfloor W/2\rfloor}\right]^T. \quad (1)$$

We use this concatenated feature vector for a new action representation, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_M) \in \mathcal{X}$, which results in substantial performance improvements in our experiments.

In general, action sequences can have variable lengths. However, they are resampled to equal length M as the preprocessing step in our method. To do this, we use the simple linear interpolation method. It is known that the resampling process does not negatively affect the performance of DTW-based sequence classification [18]. Let $C = \{1, \ldots, C\}$ be the set of all action category labels. The goal of action classification is to find a mapping function $h : \mathcal{X} \to C$.

3.1 DTW-Based Action Classification

We first briefly describe conventional DTW-based action classification approaches [7], [10] that we call DTW-NN in this paper. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$ denote an input test sequence. DTW can then be used to optimally align $\mathbf{X}$ with a training sequence $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_M)$ to produce the dissimilarity between them as follows:

$$DTW(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{P}\in\mathcal{P}} \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij} \cdot \|\mathbf{x}_i - \mathbf{y}_j\|^2, \quad (2)$$

where $\mathbf{P}$ is a $M \times M$ path matrix and $\mathcal{P} \subset \{0,1\}^{M\times M}$ is a set of all valid path matrices. The entry in the i-th row and j-th column, $P_{ij} = 1$, indicates that $\mathbf{x}_i$ is matched to $\mathbf{y}_j$. The valid path matrix $\mathbf{P}$ can be reparameterized through a path sequence $\mathbf{Q} = (\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(L)})$, where L is the length of the path and $\mathbf{q}^{(l)} = [q_x^{(l)}, q_y^{(l)}]^T$ denotes that the $q_x^{(l)}$-th frame in $\mathbf{X}$ and the $q_y^{(l)}$-th frame in $\mathbf{Y}$ are aligned. Therefore, $P_{ij} = 1$ iff $\mathbf{q}^{(l)} = [i, j]^T$ exists for some l. Here, $\mathbf{Q}$ should satisfy the following three constraints: boundary constraint ($q_x^{(1)} = q_y^{(1)} = 1$ and $q_x^{(L)} = q_y^{(L)} = M$), continuity constraint ($q_x^{(l)} - q_x^{(l-1)} \le 1$ and $q_y^{(l)} - q_y^{(l-1)} \le 1$), and monotonicity constraint ($q_x^{(l-1)} \le q_x^{(l)}$ and $q_y^{(l-1)} \le q_y^{(l)}$). The optimization problem in (2) can be solved efficiently in $O(M^2)$ time through dynamic programming [19].

Let $\mathcal{Y}_c = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(N)}\}$ be the set of N training sequences of which category label is c. We can then define the DTW between $\mathbf{X}$ and $\mathcal{Y}_c$ as follows:

$$DTW(\mathbf{X}, \mathcal{Y}_c) = \min_{n\in\{1,\ldots,N\}} DTW(\mathbf{X}, \mathbf{Y}^{(n)}). \quad (3)$$

This can be intuitively understood as the dissimilarity between an action sequence $\mathbf{X}$ and an action class c, that is, the *image-to-class distance* [20]. The input action $\mathbf{X}$ is then classified as the class with the minimum image-to-class distance as follows:

$$\hat{c} = \arg\min_{c\in C} DTW(\mathbf{X}, \mathcal{Y}_c). \quad (4)$$

The time complexity of this classification algorithm is $O(C \cdot M^2 \cdot N)$.

3.2 Constructing the Augmented Dataset

Our main claim is that the data augmentation process can be helpful for nonparametric classification approaches, such as the DTW-based approach in (4). Let us consider augmenting the training dataset $\mathcal{Y}_c$. The sequences in the set $\mathcal{Y}_c = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(N)}\}$ are combined to produce the augmented dataset $\tilde{\mathcal{Y}}_c$. Let $\tilde{\mathbf{Y}} = (\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_M)$ denote an element of $\tilde{\mathcal{Y}}_c$. We select $\tilde{\mathbf{y}}_m$ from the set of all skeletal features $\{\mathbf{y}_m^{(1)}, \ldots, \mathbf{y}_m^{(N)}\}$ at the m-th frame. Inspired by the naive Bayes classifiers [20], these selection processes are assumed to be independent of each other. As illustrated in Fig. 1, a new sequence $\tilde{\mathbf{Y}}^{(i)}$ can be created by combining M frame-level features randomly selected from N training sequences for each frame. We define a set of all sequences that can be generated through this process as the augmented dataset $\tilde{\mathcal{Y}}_c$, of which the cardinality is $|\tilde{\mathcal{Y}}_c| = N^M$. The augmented dataset $\tilde{\mathcal{Y}}_c$ can now be used for classification:

$$\hat{c} = \arg\min_{c\in C} DTW(\mathbf{X}, \tilde{\mathcal{Y}}_c). \quad (5)$$

The large cardinality $|\tilde{\mathcal{Y}}_c| = N^M$ makes the naive approach to the optimization problem in (5) infeasible with time complexity $O(C \cdot M^2 \cdot N^M)$.

In fact, various types of approaches can be considered for augmenting skeletal sequences in addition to the proposed method based on the naive Bayes assumption. For example, adding random noise to joint positions or orientations is the most common way, allowing the learned model to be robust to noise. However, this can only generate slightly modified data in which the joints of each frame are randomly perturbed from the original training sequence. On the other hand, the proposed data augmentation method can produce completely new data that cannot be generated by existing methods. For example, from two training sequences $A = abcd$ and $B = wxyz$, a new augmented sequence $C = abyz$ can be obtained. In addition, as presented in the subsequent subsection, the complexity of DTW matching using the augmented dataset generated by the proposed method can be reduced to a much smaller value than $O(C \cdot M^2 \cdot N^M)$ by our novel relaxed formulation. However, the method based on random noise increases the running time of DTW matching linearly in proportion to the number of augmented sequences.

Despite the advantages of the proposed augmentation process, it may generate unrealistic action sequences with no temporal coherence. For example, suppose an action class consists of two consecutive sub-actions, a and b. Let $A = a_1a_1a_1a_1b_1b_1$ and $B = a_2a_2b_2b_2b_2b_2$ be two sequences belonging to such a class with a length of $M = 6$. From these two sequences, the proposed augmentation method can generate an unrealistic sequence $C = a_1a_1b_2a_1b_2b_2$. This prob-

Algorithm 1 Algorithm for Pre-aligning Training Sequences

1: **Input:** a training dataset $\mathcal{Y}_c = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(N)}\}$,
the maximum number of iterations *max_iter*,
a tolerance threshold δ
2: Initialize the nominal sequence $\dot{\mathbf{Y}} \leftarrow \mathbf{Y}^{(1)}$ and $iter \leftarrow 0$
3: **while** $iter < max\_iter$ **do**
4: **for** $k = 1, \ldots, N$ **do**
5: Perform DTW between $\mathbf{Y}^{(k)}$ and the nominal sequence $\dot{\mathbf{Y}}$ to obtain a path matrix $\mathbf{P} = (P_{ij})$
6: Warp $\mathbf{Y}^{(k)}$ to the nominal sequence $\dot{\mathbf{Y}}$ using $\mathbf{P}$ to get a warped sequence $\dot{\mathbf{Y}}^{(k)} = (\dot{\mathbf{y}}_1^{(k)}, \ldots, \dot{\mathbf{y}}_M^{(k)})$:
$\dot{\mathbf{y}}_i^{(k)} \leftarrow \sum_{j=1}^{M} P_{ij}\mathbf{y}_j^{(k)} / \sum_{j=1}^{M} P_{ij}$, $i = 1, \ldots, M$
7: **end for**
8: Compute a new nominal sequence $\dot{\mathbf{Y}}' = (\dot{\mathbf{y}}'_1, \ldots, \dot{\mathbf{y}}'_M)$:
$\dot{\mathbf{y}}'_i \leftarrow \frac{1}{N}\sum_{k=1}^{N} \dot{\mathbf{y}}_i^{(k)}$, $i = 1, \ldots, M$
9: **if** $\sum_{i=1}^{M} \|\dot{\mathbf{y}}'_i - \dot{\mathbf{y}}_i\|_2^2 \leq \delta$ **then**
10: **break**
11: **end if**
12: Update $\dot{\mathbf{Y}} \leftarrow \dot{\mathbf{Y}}'$ and $iter \leftarrow iter + 1$
13: **end while**
14: **Output:** a pre-aligned dataset $\mathcal{Y}_c \leftarrow \{\dot{\mathbf{Y}}^{(1)}, \ldots, \dot{\mathbf{Y}}^{(N)}\}$

lem can be mitigated by temporally aligning the training sequences before augmentation. That is, A and B can be modified to $A' = a_1a_1a_1b_1b_1b_1$ and $B' = a_2a_2a_2b_2b_2b_2$, which prevents the above unrealistic sequence from being generated through the proposed data augmentation process.

The specific pre-aligning algorithm is summarized in Algorithm 1. The basic idea is to compute a nominal sequence for each action category [7] and then warp its corresponding training sequences to this nominal sequence using DTW. In the path matrix $\mathbf{P}$ obtained by DTW optimization of line 5, $P_{ij} = 1$ indicates that the j-th frame feature $\mathbf{y}_j^{(k)}$ of the training sequence $\mathbf{Y}^{(k)}$ is matched to the i-th frame feature $\dot{\mathbf{y}}_i$ of the nominal sequence $\dot{\mathbf{Y}}$. Since one or more frames of the training sequence may correspond to the i-th frame of the nominal sequence, the i-th frame feature $\dot{\mathbf{y}}_i^{(k)}$ of the warped sequence is defined as the average of all such features (line 6). Applying this procedure to all training sequences, we can obtain the warped training sequences aligned to the nominal sequence, which are averaged for each frame to generate a new nominal sequence $\dot{\mathbf{Y}}'$ as in line 8. The above procedure is repeated until the difference between the new and old nominal sequences is sufficiently small as in line 9. In our experiments, this pre-aligning process improves the classification performance substantially.

3.3 DTW with Augmented Dataset

We now present the proposed *data augmented dynamic time warping* (DA-DTW) algorithm, which can efficiently solve the optimization problem in (5). Let us define the *augmentation vector* $\mathbf{a} \in \mathcal{A} = \{1, \ldots, N\}^M$. One augmentation vector $\mathbf{a} = [a_1, \ldots, a_M]^T$ corresponds to a particular sequence element $(\mathbf{y}_1^{(a_1)}, \ldots, \mathbf{y}_M^{(a_M)})$ in the augmented dataset $\tilde{\mathcal{Y}}_c$. The DTW between $\mathbf{X}$ and $\tilde{\mathcal{Y}}_c$ can then be formulated as:

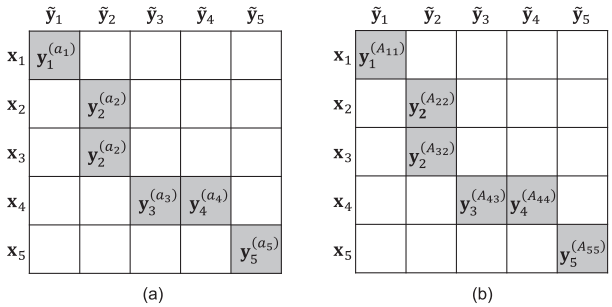


Fig. 2 An example of a path matrix is illustrated in (a), where gray and white cells denote 1 and 0, respectively. This shows the correspondences between an input sequence $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_5)$ and an augmented sequence $\tilde{\mathbf{Y}} = (\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_5)$, where the augmentation vector $\mathbf{a} = [a_1, \ldots, a_5]^T$ uniquely determines the augmented sequence as $\tilde{\mathbf{y}}_i = \mathbf{y}_i^{(a_i)}$, $i = 1, \ldots, 5$. Whereas the augmentation matrix in (b) allows ambiguities in the augmented sequence. For example, if $A_{22} \neq A_{32}$, then $\mathbf{y}_2^{(A_{22})} \neq \mathbf{y}_2^{(A_{32})}$. This leads to ambiguity in the decision of $\tilde{\mathbf{y}}_2$, and as a result prevents the augmented sequence $\tilde{\mathbf{Y}}$ from being uniquely determined.

$$DTW(\mathbf{X}, \tilde{\mathcal{Y}}_c) = \min_{\mathbf{a}\in\mathcal{A}} \min_{\mathbf{P}\in\mathcal{P}} \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij} \cdot \|\mathbf{x}_i - \mathbf{y}_j^{(a_j)}\|^2. \quad (6)$$

As discussed in Sect. 3.2, the number of all possible augmented sequences is very large, $|\mathcal{A}| = |\tilde{\mathcal{Y}}_c| = N^M$, which requires a more feasible conversion of the above optimization problem.

Our basic idea is to modify (6) by putting $\min_{\mathbf{a}}$ inside two summation operators $\sum_i \sum_j$ to make it look like (2), where the DTW algorithm based on dynamic programming can be used. However, the $\min_{\mathbf{a}}$ operator cannot be inserted in the summation $\sum_i$ as follows:

$$\begin{aligned} DTW(\mathbf{X}, \tilde{\mathcal{Y}}_c) &= \min_{\mathbf{a}\in\mathcal{A}} \min_{\mathbf{P}\in\mathcal{P}} \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij} \cdot \|\mathbf{x}_i - \mathbf{y}_j^{(a_j)}\|^2 \\ &= \min_{\mathbf{P}\in\mathcal{P}} \sum_{j=1}^{M} \min_{a_j} \sum_{i=1}^{M} P_{ij} \cdot \|\mathbf{x}_i - \mathbf{y}_j^{(a_j)}\|^2 \quad (7) \\ &\neq \min_{\mathbf{P}\in\mathcal{P}} \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij} \cdot \min_{a_j} \|\mathbf{x}_i - \mathbf{y}_j^{(a_j)}\|^2. \end{aligned}$$

For example, Fig. 2 (a) illustrates a correspondence between $\mathbf{X}$ and an augmented sequence $\tilde{\mathbf{Y}}$, where $\mathbf{y}_2^{(a_2)}$ is matched to the features $\mathbf{x}_2$ and $\mathbf{x}_3$ of two consecutive frames. This corresponds to $\min_{a_2}(\|\mathbf{x}_2 - \mathbf{y}_2^{(a_2)}\|^2 + \|\mathbf{x}_3 - \mathbf{y}_2^{(a_2)}\|^2)$ in (7), which is obviously not the same as $\min_{a_2}\|\mathbf{x}_2 - \mathbf{y}_2^{(a_2)}\|^2 + \min_{a_2}\|\mathbf{x}_3 - \mathbf{y}_2^{(a_2)}\|^2$ because, in general, $\min_x \sum_i f_i(x) \geq \sum_i \min_x f_i(x)$ holds.

To change (6) into a more manageable form, let us introduce the *augmentation matrix* $\mathbf{A} \in \{1, \ldots, N\}^{M\times M}$ and use it instead of the augmentation vector. The basic idea is to define a different variable A_{ij} for each i and j to move the min operator inside two summations $\sum_i \sum_j$ in (7). Assuming that the i-th frame $\mathbf{x}_i$ of the input sequence and the j-th frame $\tilde{\mathbf{y}}_j$ of the augmented sequence are matched, the element A_{ij} of the augmentation matrix $\mathbf{A}$ defines which training sequence

the j-th frame is selected from (i.e., $\tilde{\mathbf{y}}_j = \mathbf{y}_j^{(A_{ij})}$). Under this notation, if $P_{ij} = 1$ then $\mathbf{x}_i$ is matched to $\mathbf{y}_j^{(A_{ij})}$. However, this abuse of notation does not guarantee the uniqueness of the augmented sequence as illustrated in Fig. 2 (b), where the additional constraint, $A_{22} = A_{32}$, is needed to enforce the uniqueness. In sum, the optimization problem in (6) can be rewritten using the augmentation matrix as follows:

$$\begin{aligned} &\min_{\mathbf{A}} \min_{\mathbf{P}\in\mathcal{P}} \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij}\cdot\|\mathbf{x}_i - \mathbf{y}_j^{(A_{ij})}\|^2 \\ &\text{s.t. } \forall j : A_{1j} = A_{2j} = \ldots = A_{Mj}, \end{aligned} \tag{8}$$

where the second constraints are necessary to uniquely determine the matched training sequence $\tilde{\mathbf{Y}}$ and make the formulations of (6) and (8) the same problem. If we remove these constraints, all A_{ij}s become independent of each other, which allows $\min_{\mathbf{A}} = \min_{A_{11}} \min_{A_{12}} \ldots \min_{A_{MM}}$ to be put inside $\sum_i \sum_j$ as follows:

$$\min_{\mathbf{P}\in\mathcal{P}} \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ij}\cdot \min_{A_{ij}} \|\mathbf{x}_i - \mathbf{y}_j^{(A_{ij})}\|^2. \tag{9}$$

We propose to use the *relaxed formulation* in (9) for skeletal action classification. Note the similar structures between the relaxed problem in (9) and the original DTW problem in (2). We can efficiently solve our relaxed problem by first finding the optimal A_{ij}s for all $1 \le i, j \le M$, and then running the original DTW algorithm only once. If we use an efficient NN search algorithm, such as a kd-tree [21] to find A_{ij}, then the problem in (9) can be solved in $O(M^2 \cdot \log N)$ that is the sum of $O(M^2 \cdot \log N)$ for M^2 NN search problems and $O(M^2)$ for dynamic programming. Therefore, classifying one test sequence by (5) takes $O(C \cdot M^2 \cdot \log N)$ time. This logarithmic time complexity is a significantly better result compared with the exponential time complexity $O(C \cdot M^2 \cdot N^M)$ of the naive approach that repeats DTW matching for all N^M augmented sequences.

Surprisingly, our method has even a better time complexity than the DTW-NN method without data augmentation in (4), which has $O(C \cdot M^2 \cdot N)$. The conventional DTW-NN approach repeats DTW matching N times to calculate the image-to-class distance as in (3), whereas the proposed DA-DTW method performs DTW matching only once. Instead, in the proposed method, M^2 NN search problems should be solved before applying dynamic programming for DTW matching. In this part, the complexity can be reduced through efficient NN search algorithms such as the kd-tree method. Therefore, the algorithm speedup through the kd-tree cannot be applied to the DTW-NN method that does not include the NN search problem based on the Euclidean distance. Note that "NN" in DTW-NN means NN search with the DTW-based distance for sequences.

3.4 Handling the Multiple Modes of Actions

The proposed pre-aligning process in Sect. 3.2 can be used to find the correct nominal sequence when the distribution of action sequences has a singular mode. If there are multiple modes as illustrated in Fig. 3, the direct use of the data augmentation process could be problematic. We therefore propose to divide the training set $\mathcal{Y}_c$ into K disjoint sets $\mathcal{Y}_{c,1}, \ldots, \mathcal{Y}_{c,K}$ corresponding to multiple modes. For that purpose, we vectorize each training sequence $\mathbf{Y}^{(i)} = (\mathbf{y}_1^{(i)}, \ldots, \mathbf{y}_M^{(i)})$ belonging to the set $\mathcal{Y}_c$ and apply the k-means algorithm to the resultant vectors. After that, the proposed pre-aligning process in Sect. 3.2 and relaxed DTW method with data augmentation in Sect. 3.3 are applied to each $\mathcal{Y}_{c,k}$, $1 \le c \le C$, $1 \le k \le K$. Finally, action classification can be performed as follows:

$$\hat{c} = \arg\min_{c\in C} \min_{k\in\mathcal{K}} DTW(\mathbf{X}, \tilde{\mathcal{Y}}_{c,k}), \tag{10}$$

where $\mathcal{K} = \{1, \ldots, K\}$.

Our final algorithm for skeletal action classification is summarized in Algorithm 2. We first divide training sequences belonging to class c into K disjoint sets using the k-means algorithm (line 3). After pre-aligning each of these K sets (line 5), we construct kd-trees (line 6). The image-to-class distance between the disjoint set and the input test sequence is then calculated by solving the optimization problem in (9). To do this, we first solve the NN problem for each $1 \le i, j \le M$ using the kd-trees constructed above to determine which training sequence's j-th frame is matched to the i-th frame of the input sequence (line 10). We then solve a DTW problem defined by the M^2 distances D_{ij}, $1 \le i, j \le M$ obtained through the NN problems, which provides an image-to-class distance $S_{c,k}$ (line 12). Finally, the optimal action class $\hat{c}$ is determined by finding the class that minimizes the image-to-class distances (line 15). The k-means algorithm, pre-aligning, and building kd-trees in lines

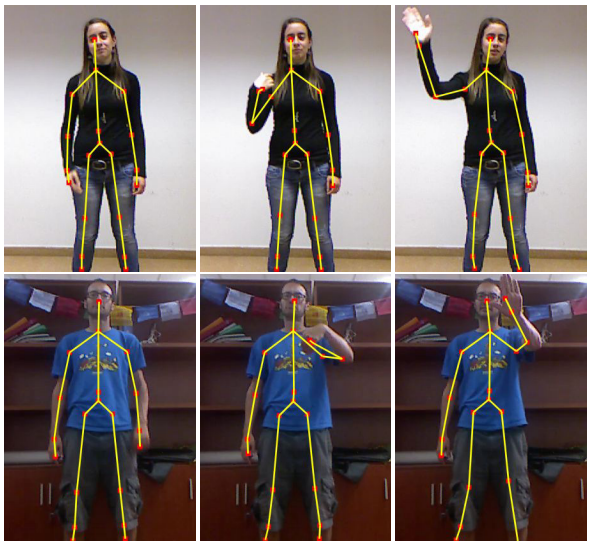


Fig. 3 Two samples of the *vattene* action in ChaLearn LAP dataset are shown. This action can be performed either the left or right hand, where the movements with different hands correspond to different modes in the distribution of the *vattene* action. Note that color images are displayed for better visualization and are not used in the proposed method.

Algorithm 2 Algorithm for Skeletal Action Classification

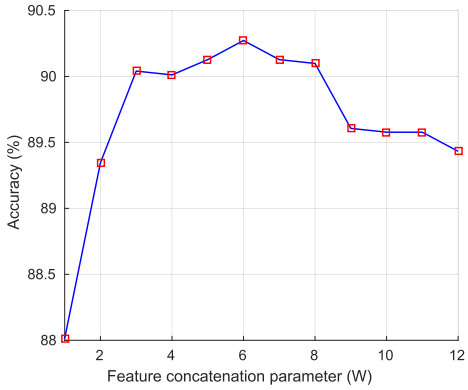
1: **Input:** a test sequence $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$,
training datasets $\mathcal{Y}_1, \ldots, \mathcal{Y}_C$,
the number of modes K
2: **for** $c = 1, \ldots, C$ **do**
3: Use k-means algorithm to split the training dataset $\mathcal{Y}_c$ into K disjoint sets: $\mathcal{Y}_{c,1}, \ldots, \mathcal{Y}_{c,K}$
4: **for** $k = 1, \ldots, K$ **do**
5: Align training sequences in $\mathcal{Y}_{c,k}$ using Algorithm 1
6: Construct kd-trees for training sequences in $\mathcal{Y}_{c,k}$
7: **end for**
8: **for** $k = 1, \ldots, K$ **do**
9: **for** $i, j = 1, \ldots, M$ **do**
10: Solve the nearest neighbor search problem using kd-trees:
$D_{ij} \leftarrow \min_{A_{ij}} \|\mathbf{x}_i - \mathbf{y}_j^{(A_{ij})}\|^2$
s.t. $\forall n \in \{1, \ldots, N\} : (\mathbf{y}_1^{(n)}, \ldots, \mathbf{y}_M^{(n)}) \in \mathcal{Y}_{c,k}$
11: **end for**
12: Solve DTW problem:
$S_{c,k} \leftarrow \min_{\mathbf{p} \in \mathcal{P}} \sum_{i=1}^{M} \sum_{j=1}^{M} P_{ij} \cdot D_{ij}$
13: **end for**
14: **end for**
15: **Output:** action class $\hat{c} \leftarrow \arg\min_c \min_k S_{c,k}$

3-7 require only training sequences. In other words, these procedures can be done offline prior to classifying a test sequence. On the other hand, the procedures in lines 8-13, which use the kd-trees to solve NN problems, perform DTW optimization, and consequently compute image-to-class distances, should be done online as a process of actually classifying a given test sequence.

4. Experimental Results

4.1 Datasets

To evaluate the performance of the proposed method, we use two different datasets for action recognition. The first dataset is the MSR-Action3D dataset [22], which consists of 20 actions (i.e., *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup & throw*) performed by ten subjects. It provides 557 action sequences in the form of depth maps and the 3D locations of 20 joints. Each sequence is assumed to contain only one action instance. We follow the experimental setting of [8], where the samples of half of the subjects are used for training, and the rest are used for testing. Random selection of the training and test subjects is repeated ten times, generating ten datasets for the evaluation. The second is the ChaLearn LAP dataset introduced during the 2014 ChaLearn Looking at People (LAP) Challenge [23]. The target actions include 20 Italian cultural/anthropological hand gestures (i.e., *vattene, vieniqui, perfetto, furbo, cheduepalle, chevuoi, daccordo, seipazzo, combinato, freganiente, ok, cosatifarei, basta, prendere, noncenepiu, fame, tantotempo, buonissimo, messidaccordo, sonostufo*). The dataset comprises 940 sequences (470 training, 230 validation, and 240 test sequences), where each sequence contains RGB, depth map, and 3D joint information. A total of 13,880 action instances (6,847 training, 3,454 validation, and 3,579 test instances) exist, the scale of which is significantly larger than that of the MSR-Action3D dataset. Note that the actions used in our experiments consist of the relatively simple movements of body parts, not complex behaviors such as cooking and sweeping.



(a)

Fig. 4 The cross validation process for determining the optimal feature concatenation parameter W is illustrated for the ChaLearn LAP dataset. The accuracies of the proposed method are computed with a varying W for the validation set. The optimal W is determined to be 6 in this case.

4.2 Implementation Details

From the dataset, we discard the RGB and depth data, and use only the skeletal joint data to compute the feature representation defined in (1). For the MSR-Action3D dataset, the 20 joints of the full body are first normalized using the reference hip center joint, and then utilized for the feature computation. In the case of the ChaLearn LAP dataset, we consider only the ten joints of the upper body (that is, the head, neck, L/R shoulder, L/R elbow, L/R wrist, and L/R hand) with the reference neck joint because they are sufficient to express all hand gestures in the dataset. The sequence length parameter for the MSR-Action3D dataset is set to $M = 76$, which is same as the experimental setting in [7]. For the large-scale ChaLearn LAP dataset, we use a relatively small number of $M = 32$ frames to reduce the computational burden. The feature concatenation parameter W and the number of modes K are determined through cross validation. The cross validation process for determining W for the ChaLearn LAP dataset is illustrated in Fig. 4, where the difference in performance due to the change of W is not large and this shows that the proposed method is robust to the choice of W. The open-source VLFeat library [24] is used for an efficient NN search with a kd-tree, in which the search speed can be controlled by changing the upper bound κ on the number of points for examination. To compare the performance of the proposed method with other well-known standard classifiers, such as the k-nearest neighbors (kNN), support vector machines with a radial basis function kernel (SVM-RBF), and random forests (RF), we use the open-

Table 1 Classification accuracies for MSR-Action3D dataset.

Method	Accuracy (%)
kNN	71.70
DTW-NN	84.25
DA-DTW ($K = 1$) without pre-aligning	85.79
DA-DTW ($K = 1$) with pre-aligning	**86.82**
SVM-RBF	75.43
RF	73.45
Multiple Instance Learning [5]	65.70
Structured Streaming Skeletons [12]	81.70
DBN-HMM [9]	82.00
Actionlet Ensemble [8]	88.20
Lie Group [7]	89.48

source Scikit-learn library [25]. Their hyper-parameters are obtained through cross validation.

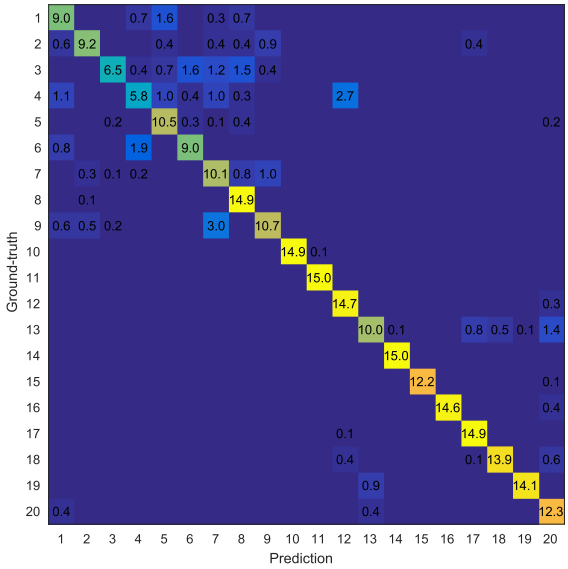
4.3 Performance Analysis on the MSR-Action3D Dataset

The evaluation results of the proposed action classification method on the MSR-Action3D dataset are presented. We first show that the proposed DA-DTW method improves the performance of the conventional DTW-based NN classification approach [10]. Table 1 shows the performances of the various nonparametric approaches, including ours. First, the DTW-based approaches significantly outperform the simple kNN method, which shows that the robustness to the speed variation plays a crucial role in action classification. The proposed DA-DTW method produces substantially improved results compared with the DTW-NN. The best accuracy (86.82%) is achieved by aligning the training dataset with the nominal sequences before performing relaxed DTW matching, which shows the effectiveness of the pre-aligning process.

Comparative results with other learning-based skeletal action classification methods are also shown in Table 1. Our method outperforms both SVM-RBF and RF classifiers, where all frame-level features $\mathbf{z}_m, 1 \leq m \leq M$ are concatenated into a single feature vector. Most of the competitive approaches in the literature are based on learning processes, such as multiple instance learning [5], deep belief networks [9], multiple kernel learning [8], and linear support vector machines [7]. Although the proposed method depends on a simple nonparametric matching process without learning, it still achieves a comparable performance with other state-of-the-art methods. The confusion matrix of the proposed method is shown in Fig. 5. We can see that the accuracies of the third and fourth action classes (i.e., *hammer* and *hand catch*) are relatively low. Both of these actions involve interactions between human and objects, which are hardly represented by human skeletal information alone.

4.4 Performance Analysis on the ChaLearn LAP Dataset

In this subsection, we present the evaluation results for the large-scale ChaLearn LAP dataset. The publicly available code of the state-of-the-art method in [7] is used for comparison. Table 2 shows that the pre-aligning process does



(a)

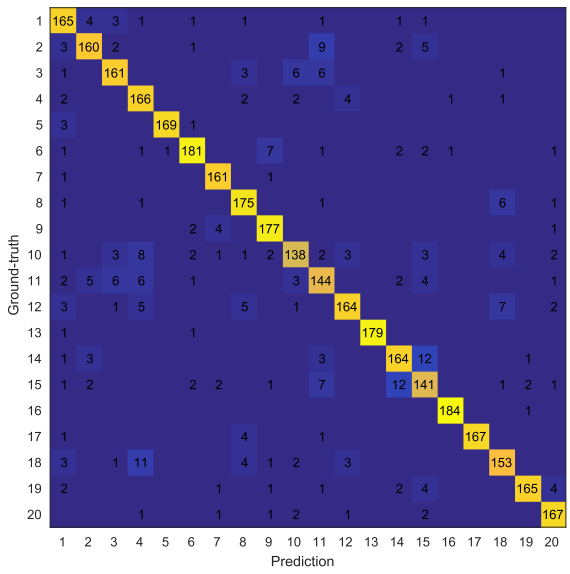
Fig. 5 The confusion matrices for ten random splits of training/test samples are averaged for MSR-Action3D dataset. The numbers in the i-th row and j-th column indicate the number of samples that belong to action category i but are classified as category j.

Table 2 Classification accuracies for ChaLearn LAP dataset.

Method	Accuracy (%)
kNN	85.55
DTW-NN	89.10
DA-DTW ($K = 1$) without pre-aligning	91.42
DA-DTW ($K = 1$) with pre-aligning	91.37
DA-DTW ($K = 2$) with pre-aligning	**91.56**
SVM-RBF	87.12
RF	88.54
Lie Group [7]	88.64

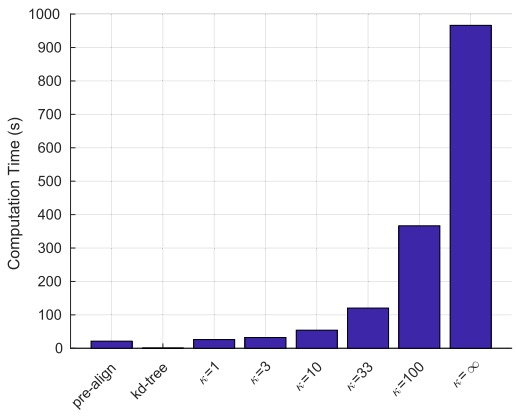
not improve the performances under the assumption of a singular mode ($K = 1$). This is because the gestures of the ChaLearn LAP dataset contain multiple modes. We can see that setting $K = 2$ makes the pre-aligning process effective. The proposed method produces the best result. Surprisingly, nonparametric approaches, including the simple kNN method, generally perform well compared with the learning-based approaches. This result shows that the nonparametric approaches, including the proposed method, can significantly benefit from the large-scale training data. Figure 6 illustrates the confusion matrix of the proposed method. Accuracies are very high for most actions, indicating that the proposed method is effective for recognizing relatively simple gestures given a large dataset.

We investigated the computational complexity of the proposed method. To do so, a 3.5 GHz 6-core CPU machine with 128 GB of RAM was utilized. The proposed method can be divided into two steps: a *training process* in which training sequences are pre-aligned and kd-trees for all action classes are constructed, and a *test process* in which action recognition is performed online for a given test sequence. Figure 7 shows the computation times for the training and



(a)

Fig. 6 The confusion matrix for ChaLearn LAP dataset.

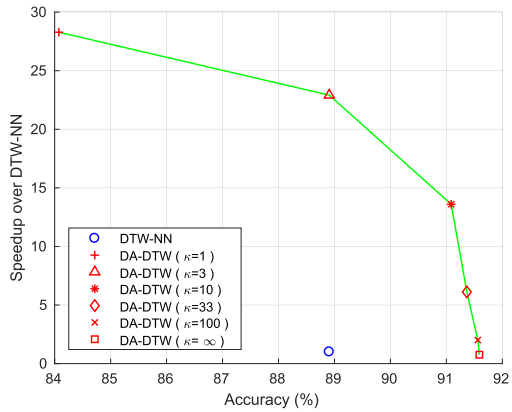


(a)

Fig. 7 The computation times of the proposed method (DA-DTW) are displayed. The first and second bars represent the times for pre-aligning the training sequences and building kd-trees, respectively. The remaining bars represent the proposed DTW matching times for the test sequences with a varying parameter κ of the kd-tree.

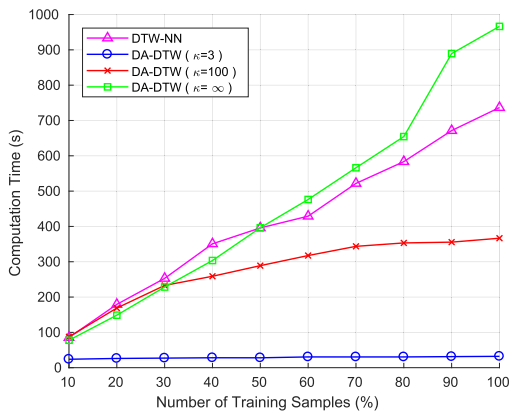
test steps of the proposed DA-DTW method. The computation time for training with a total of 10,301 training and validation sequences was approximately 22 seconds, which is reasonable. The online action recognition was performed for a total of 3,579 test sequences and classifying one test sequence took 7, 9, 15, 33, 102, and 270 milliseconds for $\kappa = 1$, 3, 10, 33, 100, and ∞, respectively. Since the training step can be accomplished offline, we consider only the test step in the subsequent complexity analysis.

One of our main claims is that our DA-DTW with augmented training sequences can perform even faster than the conventional DTW-NN without augmentation. Moreover, the tradeoff between accuracy and speed can be controlled by changing the parameter κ of the kd-tree in the proposed

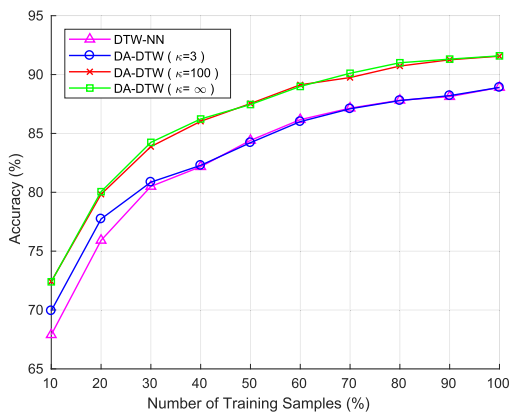


(a)

Fig. 8 The performances of the DTW-NN and the proposed method (DA-DTW) are illustrated with a varying upper bound (κ) on the number of examination points in the kd-tree. $\kappa = \infty$ corresponds to the exact NN search.



(a)



(b)

Fig. 9 The computation times and accuracies of the DTW-NN and the proposed method (DA-DTW) are plotted as a function of the different sizes of the training dataset. The ChaLearn LAP dataset is used for these experiments.

method. This result is illustrated in Fig. 8. With $\kappa = 3$, the proposed method runs about 23-times faster than the DTW-NN, but the performance of the proposed method is on par

with that of the DTW-NN. To investigate the scalability of our method, we conducted experiments with a varying number of training sequences. The computation times and accuracies are illustrated in Fig. 9, which illustrates that the proposed DA-DTW with a proper $\kappa = 100$ works very efficiently without a sacrifice in accuracy.

5. Conclusion

We proposed a novel DTW-based method for skeletal action classification. Our approach is based on data augmentation to improve the generalization power and relaxed DTW formulation for runtime efficiency. Our simple and intuitive nonparametric approach achieves a recognition performance comparable with other sophisticated learning-based methods. Moreover, through its learning-free nonparametric nature, the high flexibility of the proposed method allows additional training sequences or even new action categories to be easily added to the dataset. In the future, we aim to apply our approach to a more practical real-time action detection problem.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; Ministry of Science, ICT & Future Planning) (No. 2017R1C1B5017371) and the Research Grant of Kwangwoon University in 2018.

References

[1] J.K. Aggarwal and M.S. Ryoo, "Human activity analysis: A review," ACM Computing Surveys, vol.43, no.3, pp.16:1–16:43, 2011.
[2] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall, "A survey on human motion analysis from depth data," Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, vol.8200, pp.149–187, Springer Berlin Heidelberg, 2013.
[3] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," IEEE Transactions on Cybernetics, vol.43, no.5, pp.1318–1334, 2013.
[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.1297–1304, 2011.
[5] C. Ellis, S.Z. Masood, M.F. Tappen, J.J. LaViola, and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition," International Journal of Computer Vision, vol.101, no.3, pp.420–436, 2013.
[6] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.1737–1746, 2012.
[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.588–595, 2014.
[8] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.1290–1297, 2012.
[9] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.724–731, 2014.
[10] J. Wu, J. Cheng, C. Zhao, and H. Lu, "Fusing multi-modal features for gesture recognition," Proceedings of the 15th ACM on International Conference on Multimodal Interaction, pp.453–460, 2013.
[11] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," Proc. IEEE Int'l Conf. Computer Vision (ICCV), pp.2752–2759, 2013.
[12] X. Zhao, X. Li, C. Pang, X. Zhu, and Q.Z. Sheng, "Online human gesture recognition from motion data streams," Proceedings of the 21st ACM International Conference on Multimedia, pp.23–32, 2013.
[13] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M.J. Black, "Towards understanding action recognition," Proc. IEEE Int'l Conf. Computer Vision (ICCV), pp.3192–3199, 2013.
[14] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Transactions on Acoustics, Speech and Signal Processing, vol.26, no.1, pp.43–49, 1978.
[15] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp.1097–1105, 2012.
[16] G. Rogez and C. Schmid, "Mocap-guided data augmentation for 3d pose estimation in the wild," Advances in Neural Information Processing Systems, pp.3108–3116, 2016.
[17] T. Devries and G. Taylor, "Dataset augmentation in feature space," International Conference on Learning Representations (ICLR) Workshop Track, 2017.
[18] C.A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," Proceedings of SIAM International Conference on Data Mining, pp.506–510, 2005.
[19] D.P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific Belmont, Massachusetts, 1996.
[20] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.1–8, 2008.
[21] M. Muja and D.G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.36, no.11, pp.2227–2240, 2014.
[22] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," Computer Vision and Pattern Recognition Workshops (CVPRW), pp.9–14, 2010.
[23] S. Escalera, X. Baró, J. Gonzàlez, M.A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H.J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," Proc. European Conf. Computer Vision Workshops (ECCVW), vol.8925, pp.459–473, 2014.
[24] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.
[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol.12, pp.2825–2830, 2011.

Ju Yong Chang received the B.S. and Ph.D. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2001 and 2008, respectively. From 2008 to 2009, he was a postdoctoral researcher at Mitsubishi Electric Research Lab. (MERL), Cambridge, MA. Currently, he is an associate professor at Kwangwoon University, Seoul, Korea. His research interests include computer vision and machine learning, with a focus on human body pose estimation and human activity recognition.

Yong Seok Heo received the BS degree in Electrical Engineering in 2005, and the MS and the Ph.D. degrees in Electrical Engineering and Computer Science in 2007 and 2012, respectively, from Seoul National University, Korea. During 2012 to 2014, he was with Samsung Electronics, in the Digital Media and Communications R&D Center. Currently, he is with the Department of Electrical and Computer Engineering at Ajou University as an assistant professor. His research interests include segmentation, stereo matching, 3D reconstruction, and computational photography.