Enhanced Sequence Matching for Action Recognition from 3D Skeletal Data

Hyun-Joo Jung and Ki-Sang Hong

San 31 Hyojadong Pohang, South Korea POSTECH E.E.
Image Information Processing Lab.

Abstract. Human action recognition using 3D skeletal data has become popular topic with the emergence of the cost-effective depth sensors, such as Microsoft Kinect. However, noisy joint position and speed variation between actors make action recognition from 3D joint positions difficult. To address these problems, this paper proposes a novel framework, called Enhanced Sequence Matching (ESM), to align and compare action sequences. Inspired by DNA sequence alignment method used in bioinformatics, we model the new scoring function to measure the similarity between two action sequences with noise. We construct action sequence from a set of elementary Moving Poses (eMP) built from affinity propagation. By using affinity propagation, eMP set is built automatically, in other words, it determines the number of eMPs itself. The proposed framework outperforms the state-of-the-art on UTKinect action dataset and MSRC-12 gesture dataset and achieves comparable performance to the state-of-the-art on MSR action 3D dataset. Moreover, experimental results show that our method is very intuitive and robust to noise and temporal variation.

1 Introduction

Human action recognition has been an important area in computer vision due to its wide range of applications such as surveillance systems, human-computer interactions, and video analysis. While many existing recognition approaches achieve good results, recognizing human action from the RGB video still remains a challenging problem because it cannot fully capture the 3D human motion and is highly sensitive to illumination change or background clutter.

The recent introduction of the cost-effective depth sensors alleviates these problems. Specifically, 3D human skeletal data extracted from depth video enriches the motion information and is insensitive to the illumination change or background clutter. Estimating the skeletal joint positions from a single depth image [1] stimulates a renewed interest in skeleton-based action recognition.

In skeleton-based approaches, human action is considered as a *temporal evolution* of joint configurations and the position of each joint is considered as function of time. Therefore, modeling joint configurations and temporal evolution of actions are important tasks. However, estimated joint positions often have flipped noise in short-duration (usually less than 1 second) because of the noisy

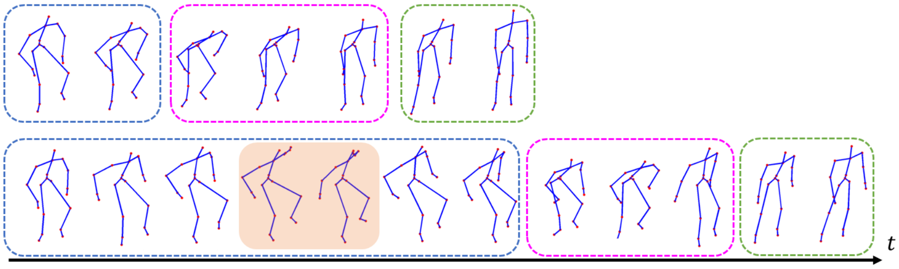

Fig. 1. Two example action sequences in the *stand up* action class of UTKinect action dataset. In both sequences, same-colored dashed box represents the same phase of the action and orange-colored box indicates that the noisy joint positions exist (in the left arm). As we can see, there are severe action speed variation and noisy poses even though both actors perform the same action.

property of depth data. Moreover, modeling temporal evolution of actions with this unstable joint positions is a difficult task. Fig. 1 shows an example of two action sequences in the same action class but have different action speed and noisy joint positions.

In this paper, to address these problems, we propose a new framework for human action recognition with 3D skeletal data which is called Enhanced Sequence Matching (ESM). Inspired by DNA sequence alignment approach used in bioinformatics [2–5], we construct action sequence from a set of elementary Moving Poses (eMPs) and match two sequences using a new scoring function. In DNA sequence alignment, if two sequences of DNA share a common ancestor, mismatches can be interpreted as point mutations and gaps as indels. Likewise, we consider human actions as a sequence of eMPs and assume that if two action sequences come from the same action class, mismatching eMPs in short-duration can be considered as a noise and should not influence the recognition result. To deal with such short-duration mismatch (mutation or gap in DNA sequence), we apply the *affine gap penalty* to our scoring function. Moreover, we construct the set of eMPs automatically by using affinity propagation [6]. Experimental results show the effectiveness of our approach, achieving the best results on UTKinect dataset and MSRC-12 gesture dataset and comparable results to the state-of-the-art on MSR action 3D dataset.

This paper is organized as follows. Section 2 reviews the related work. Our action recognition framework, elementary Moving Pose construction and Enhanced Sequence Matching (ESM), is explained in Section 3. Section 4 shows experimental results. Conclusion is in section 5.

2 Related Work

In this section, we briefly review various skeleton-based human action recognition approaches and modeling temporal evolution of human actions.

Human action recognition using 3D skeletal data has been an active area of research for the past few years. Wang et al. [7] select an informative subset of joints as an actionlet and classifies actions using ensemble of actionlet. Their actionlet is robust to noise occurring in uninformative joints and to intra-class variation. Xia et al. [8] represent human poses as a histogram of joint locations in spherical coordinate for view-invariance. Zanfir et al. [9] propose a new descriptor for 3D skeletal data named as Moving Pose (MP) descriptor which includes both pose information and differential (speed and acceleration) information. MP descriptor is invariant to scale change and absolute speed of actions. Wang et al. [10] group the joints into five body parts and uses data mining to obtain a spatial-temporal configurations of human actions. As the first step, they improve the method that estimates human joint locations to reduce errors that comes from wrongly estimated joint positions. Luo et al. [11] construct a dictionary of poses with group sparsity and geometry constraints. By adding these constraints, learned dictionary is robust to noise and large intra-class variations. As we can see, most of skeleton-based action recognition approaches mainly focus on dealing with noise and intra-class variation such as action speed change. Our method also considers the noise and speed variation by transforming an action sequence into refined action sequence and using a novel sequence matching method.

There have been many approaches for modeling temporal evolution of actions. The simplest method is using temporal pyramid [11, 7]. Luo et al. [11] divide an action sequence into 3 levels with each level containing 1, 2, 4 segments. Then histograms of sparse coefficients are generated from each segment by max pooling and concatenated to form the representation of the action sequence. Wang et al. [7] use Fourier temporal pyramid as a representation of temporal structure. For each segment at each pyramid level, they apply short time Fourier transform, obtain Fourier coefficients, and utilize its low-frequency coefficients as features. Temporal pyramid approach is easy to use and can be combined with various classification schemes such as variant of Support Vector Machine (SVM) [12, 13]. However, it only works properly when the action sequence is well segmented.

Another methods employ generative model [14, 15]. Hidden Markov Model (HMM) [16–18, 8] and Conditional Random Field (CRF) [19–21] are popular models for this approach. These methods attempt to model the generative process of actions and perform learning and inference for recognizing actions. It produces effective representation of action because it exploits the structural information of actions but learning generative model with limited amount of training data is prone to overfit.

The most similar model to our approach is temporal warping [22–25]. Dynamic Time Warping (DTW) is the most popular model for temporal warping. Veeraraghavan and Roy-chowdhury [22] compute a nominal activity trajectory for each action category using DTW. Müller and Röder [23] use DTW for matching the 3D joint positions to a motion template. Wang and Wu [25] unify DTW and SVM using Maximum Margin Temporal Warping (MMTW). The fundamental purpose of DTW is to align action sequences and to find the best warping

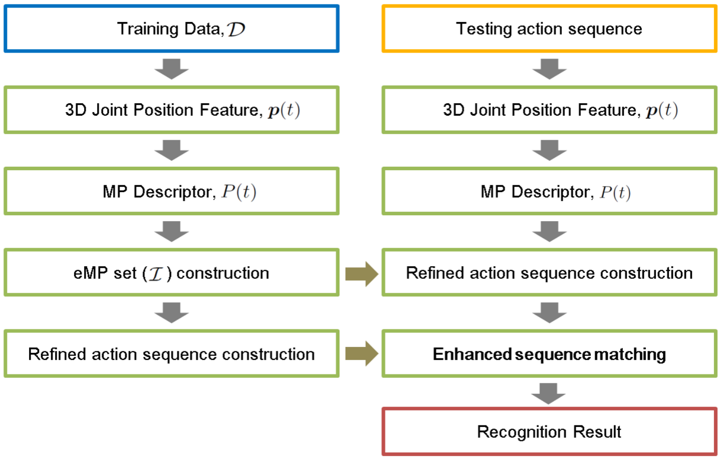
Training Data, $\mathcal{D}$
3D Joint Position Feature, $\boldsymbol{p}(t)$
MP Descriptor, $P(t)$
eMP set ($\mathcal{I}$) construction
Refined action sequence construction
Testing action sequence
3D Joint Position Feature, $\boldsymbol{p}(t)$
MP Descriptor, $P(t)$
Refined action sequence construction
Enhanced sequence matching
Recognition Result

Fig. 2. Flow chart of the proposed system.

path between two sequences. Because it focuses on measuring similarity between individual elements, it suffers severely from noisy elements in the sequence which occurs frequently in the skeletal data.

Our method mainly focus on matching sequences including noisy elements for the classification rather than finding optimal warping path. Therefore our method considers matching with the *gap* rather than matching with the noisy element in comparing sequences. Fig. 2 shows the entire system of our method.

3 Proposed Method

3.1 Feature Extraction

We use Moving Pose(MP) descriptor [9] as a feature of each frame. The position of joint j at frame t is defined by $p^j(t) = (p^j_x(t), p^j_y(t), p^j_z(t))$, where $j \in \{1, ..., J\}$ and J is the number of joints. At frame t, the human pose $\boldsymbol{p}(t)$ is represented as

$$\boldsymbol{p}(t) = \{p^j(t) | j \in \{1, ..., J\}\} \in \mathbb{R}^{J \times 3}, \tag{1}$$

namely, the concatenation of all joints' positions.

We normalize $\boldsymbol{p}(t)$ like [9]. First we compute the expected length of skeleton limbs from the training data, and modify each joint's location so that all subjects have the same limb length. After that, we subtract the position of hip center from each joint position in order that the human pose is invariant to locations.

Before computing the MP descriptor, we apply the 5-tap 1D Gaussian filter($\sigma = 1$) to each coordinate of the normalized pose along the temporal axis. Given the normalized and filtered pose $\tilde{\boldsymbol{p}}(t)$, the MP descriptor $P(t)$ is defined as

$$P(t) = (\tilde{\boldsymbol{p}}(t), \alpha\tilde{\boldsymbol{p}}'(t), \beta\tilde{\boldsymbol{p}}''(t)) \in \mathbb{R}^{J \times 3 \times 3}, \tag{2}$$

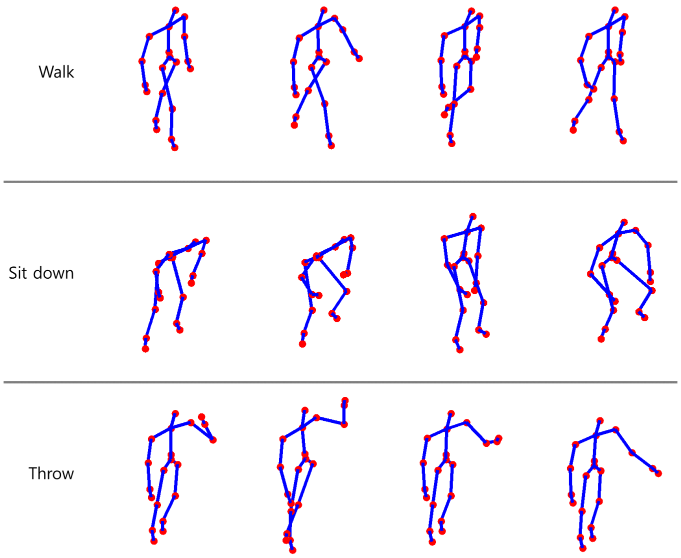

Fig. 3. Example of elementary Moving Poses of *Walk, Sit down, and Throw* action classes in the UTKinect action dataset. Note that we only plot the positional part of MP descriptor not the two derivatives.

where $\tilde{\boldsymbol{p}}'(t) = \tilde{\boldsymbol{p}}(t+1) - \tilde{\boldsymbol{p}}(t-1)$ and $\tilde{\boldsymbol{p}}''(t) = \tilde{\boldsymbol{p}}(t+2) + \tilde{\boldsymbol{p}}(t-2) - 2\tilde{\boldsymbol{p}}(t)$ are the first and second order derivatives of $\tilde{\boldsymbol{p}}(t)$ respectively. $\tilde{\boldsymbol{p}}'(t)$ and $\tilde{\boldsymbol{p}}''(t)$ are normalized so that they have unit-norm. α and β are the parameters that weight the relative importance of the two derivatives. The MP descriptor is a good feature for action classification because it captures both the static pose information and the joint kinematics at a given time.

3.2 Elementary Moving Pose (eMP)

We denote the training dataset by $\mathcal{D} = \{(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)\}$ where $X_n = \{P_n(t) | t = 1, ..., T_n\}$ is the n-th action sequence and T_n is its total frame number. $Y_n \in \mathcal{Y}$ is the action category label of n-th action sequence and $\mathcal{Y}$ is the set of action categories in the training dataset $\mathcal{D}$.

Let the set of entire MP descriptors in $\mathcal{D}$ is $\mathcal{P} = \{P_i | i = 1, ..., N_{\mathcal{D}}\}$ and its corresponding action category label set is $\{y_i | i = 1, ..., N_{\mathcal{D}}\}$ where $N_{\mathcal{D}} = \sum_{n=1}^{N} T_n$ is the total number of MP descriptors in the training set $\mathcal{D}$. We compute the class-confidence value $V(P_i)$ of P_i which is defined as

$$V(P_i) = \frac{\kappa_{y_i}}{\kappa}\,, \tag{3}$$

where κ is the number of nearest MP descriptors which is defined by user and κ_{y_i} is the number of MP descriptors that have the same action category label as P_i [9]. Large class-confidence value $V(P_i)$ means that P_i strongly belongs to its class y_i and consequently P_i has discriminative power. By thresholding class-confidence value, we exclude MP descriptors that have low discriminative power. Then the candidate of eMP is defined as

$$\tilde{\mathcal{P}} = \left\{ P_i | V(P_i) > \tau(y_i) \right\}, \tag{4}$$

where the $\tau(y_i)$ is a class-specific threshold.

To construct the eMP set, we use affinity propagation [6] which clusters features by selecting representative features as exemplars (cluster centers). The input of affinity propagation is a $N_{\tilde{\mathcal{P}}} \times N_{\tilde{\mathcal{P}}}$ affinity matrix S, where $N_{\tilde{\mathcal{P}}}$ is the number of MP descriptors in $\tilde{\mathcal{P}}$. S is defined as

$$S(i,k) = -\|P_i - P_k\|^2, \qquad \forall i \neq k \tag{5}$$

$$S(k,k) = V(P_k) - \nu, \tag{6}$$

where ν is a parameter which controls the number of exemplars. Each off-diagonal element of S encodes affinity between two MP descriptors and the k-th diagonal elements give *preference* for choosing the k-th MP descriptor as an exemplar. Equ. (6) means that the MP descriptor with large class confidence value is more likely to be chosen as an exemplar. Then, the selected MP descriptors have both representativeness as well as the discriminative power.

After the matrix S is computed, then the affinity propagation method iteratively updates the responsibility $r(i,k)$ and availability $a(i,k)$ of all pairs of data [6]. The output of affinity propagation is a set of exemplars $\mathcal{E}$ denoted by

$$\mathcal{E} = \left\{ P_i | r(i,i) > 0 \right\}, \tag{7}$$

where $r(i,i)$ is the self-responsibility of P_i. The main advantage of affinity propagation is that we do not need to specify the number of exemplars and we can assign the potential for selecting as an exemplar to each P_i. We call each exemplar as an *elementary Moving Pose* (eMP). Fig. 3 shows the example of constructed eMP. As we can see, the constructed eMP is representative of each action class and discriminative between different action classes.

Then we can rewrite $\mathcal{E}$ as

$$\mathcal{I} = \left\{ I(m) | m = 1, ..., N_{\mathcal{I}} \right\}, \tag{8}$$

where $I(m)$ is the m-th eMP(exemplar) and $N_{\mathcal{I}}$ is the number of eMPs. In the next subsection, we transform the action sequence into *refined* action sequence using $\mathcal{I}$.

3.3 Refined Action Sequence

Given an action sequence $X = \{P(t)|t = 1, ..., T\}$, we can transform X into *refined* action sequence R_X using $\mathcal{I}$. For each frame t, the distances between

$P(t)$ and $I(m)\,(m = 1, ..., N_{\mathcal{I}})$ are computed and the closest eMP is matched to the $P(t)$. We denote the index of matched eMP as

$$M(P(t)) = \min_{m} \|P(t) - I(m)\|^2 \,. \tag{9}$$

If the closest distance is larger than a pre-defined threshold ρ, then we consider $P(t)$ as a noisy frame because there is no similar eMP in the learned eMP set $\mathcal{I}$. In this case, we exclude the matched eMP from the refined action sequence.

$$M(P(t)) = 0, \qquad \text{if } \|P(t) - I(m)\|^2 > \rho, \, \forall m \in \{1, ..., N_{\mathcal{I}}\} \,. \tag{10}$$

Then the refined action sequence of X is represented as $\tilde{R}_X$.

$$\tilde{R}_X = \{M(P(t)) | M(P(t)) \neq 0, t = 1..., T\} \,. \tag{11}$$

To make the refined action sequence compact, if the same eMP is matched continuously, then we merge those frames into one element. Then the final refined action sequence becomes

$$R_X = \mathcal{U}(\tilde{R}_X) \,, \tag{12}$$

where the function $\mathcal{U}(\cdot)$ merges the continuous same value into one value. For example, $\mathcal{U}(\{5, 5, 5, 1, 7, 8, 8\}) = \{5, 1, 7, 8\}$.

3.4 Enhanced Sequence Matching (ESM)

Before explaining ESM, we mention about the DTW and traditional sequence alignment (SA) method.

Given the two action sequences X_1 and X_2, the cost of DTW is computed as

$$F_{DTW}(i, j) = D(P_1(i), P_2(j)) + \min \begin{cases} F_{DTW}(i-1, j-1) \\ F_{DTW}(i-1, j) \\ F_{DTW}(i, j-1) \end{cases} , \tag{13}$$

where $P_1(i)$ and $P_2(j)$ are MP descriptors of X_1 at frame i and X_2 at frame j respectively. $D(P_1(i), P_2(j))$ is a matching cost and the Euclidean distance is used in general. Even though DTW is very effective method for aligning two sequences, the matching cost always increase except that $P_1(i)$ and $P_2(j)$ are same. In other words, when noisy MP exists in the sequence, the cost will grow rapidly because the matching cost with noisy MP is usually large. It would degrade the classification performance.

SA [3, 4] used in bioinformatics computes the alignment score as

$$F_{SA}(i, j) = \max \begin{cases} F_{SA}(i-1, j-1) + H(R_{X1}[i], R_{X2}[j]) \\ F_{SA}(i-1, j) - \zeta \\ F_{SA}(i, j-1) - \zeta \end{cases} , \tag{14}$$

where ζ is a gap penalty parameter and $R_{X1}[i]$ and $R_{X2}[j]$ are the i-th element of refined action sequence R_{X1} and j-th element of R_{X2} respectively. $H(R_{X1}[i], R_{X2}[j])$ is the matching score written as

$$H(R_{X1}[i], R_{X2}[j]) = \begin{cases} \omega & \text{if } R_{X1}[i] = R_{X2}[j] \\ \delta & \text{if } R_{X1}[i] \neq R_{X2}[j] \end{cases} , \tag{15}$$

where the parameter $\omega > 0$ is a matching reward and $\delta < 0$ is a mismatching cost. As we can see in Equ. (14), SA considers matching each element in a sequence with both an element in another sequence and gap. However, SA gives the same matching score or mismatching cost regardless of similarity between $I(R_{X1}[i])$ and $I(R_{X2}[j])$. SA also gives the same gap penalty without regard to the length of matching with gap.

Our proposed method, named as Enhanced Sequence Matching (ESM), computes the alignment score as

$$F_{ESM}(i,j) = \max \begin{cases} F_{ESM}(i-1,j-1) + S(R_{X1}[i]), R_{X2}[j]) \\ \max_{k=0,\dots,i-1} F_{ESM}(k,j) - \gamma(|j-k|) \\ \max_{k=0,\dots,j-1} F_{ESM}(i,k) - \gamma(|i-k|) \end{cases} , \qquad (16)$$

where $S(R_{X1}[i], R_{X2}[j])$ is a matching score and $\gamma(n)$ is an *affine gap function* [5] that enables our method to model the desired property for skeleton-based action recognition. The matching score $S(R_{X1}[i], R_{X2}[j])$ is defined as

$$S(R_{X1}[i], R_{X2}[j]) = \lambda * S_{app}(I(R_{X1}[i]), I(R_{X2}[j])) + (1-\lambda) * S_{hist}(R_{X1}[i], R_{X2}[j]) , \qquad (17)$$

where λ is a parameter that controls the weights of the two similarity functions. $S_{app}(I(R_{X1}[i]), I(R_{X2}[j]))$ is the appearance similarity defined as

$$S_{app}(I(R_{X1}[i]), I(R_{X2}[j])) = \phi\left(\|I(R_{X1}[i]) - I(R_{X2}[j])\|^2\right) , \qquad (18)$$

where $\phi(\cdot)$ is a sigmoid-like function defined as $\phi(x) = \frac{1}{x+1/2} - 1$ and the class-distribution similarity, $S_{hist}(R_{X1}[i], R_{X2}[j])$, is defined as

$$S_{hist}(R_{X1}[i], R_{X2}[j]) = \phi\left(\|h(R_{X1}[i]) - h(R_{X2}[j])\|^2\right) , \qquad (19)$$

where $h(R_{X1}[i])$ is the class-distribution of $R_{X1}[i]$-th eMP which is computed from the training dataset. The b-th bin of $h(R_{X1}[i])$ is defined as

$$h^b(R_{X1}[i]) = \frac{1}{N} \sum_{n:Y_n=b} \sum_{t=1}^{T_n} \delta(R_{X1}[i], R_{Xn}[t]) , \qquad (20)$$

where N is the number of total eMPs of refined action sequence in the training set and $\delta(m_1, m_2) = 1$ if $m_1 = m_2$ and 0 otherwise. By considering class distribution of each eMP, we can additionally give more weight to the eMP that frequently occurs in the specific action class.

The affine gap function $\gamma(n)$ is defined as

$$\gamma(n) = \max[(n-1) * \eta - \theta, 0] , \qquad (21)$$

where η and θ are affine-cost and gap-cost parameter respectively. By using the affine gap function, we can ignore the matching with gap up to $\theta/\eta + 1$ elements for computing matching score because this gap-matching is down to noise. But we impose a gap penalty to matching with gap more than $\theta/\eta + 1$ elements since

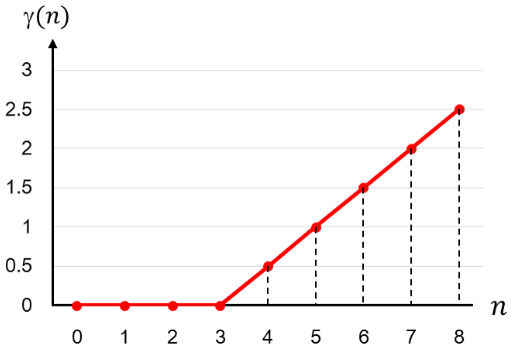

Fig. 4. Example of affine gap function when $\eta = \frac{1}{2}$ and $\theta = 1$. In this case, as we can see, we ignore the matching with gap up to 3 elements.

this gap-matching is thought to be caused by different action classes. Fig. 4 shows an example of affine gap function. This property is desirable for action sequence matching with noise, especially using 3D skeleton data for action classification. Moreover, our sequence matching approach is very intuitive and natural then other temporal modeling such as temporal pyramid. In the next section, we show that the effectiveness of our framework.

4 Experimental Results

We use MSR action 3D dataset [18], UTKinect action dataset [8], and MSRC-12 gesture dataset [26] to evaluate our proposed action classification framework.

In all datasets, there are $J = 20$ joints (head, shoulder center, shoulder left/right, elbow left/right, wrist left/right, hand left/right, spine, hip center, hip left/right, knee left/right, ankle left/right, foot left/right) to represent the human pose. For MP descriptor, we set $\alpha = 0.75$ and $\beta = 0.6$ which is the same as [9]. In the following experiments, unless specified, we use $\kappa = 50$, $\lambda = 0.7$, and $\theta = 1$. ν in Equ. (6) is determined so that the average number of eMPs for each action class is around 50. For classification, we use the nearest-neighbor scheme. The matching score of two refined action sequences is defined by the maximum value of $F_{ESM}(i, j)$ in Equ. (16).

4.1 Datasets

MSR Action 3D Dataset. MSR action 3D dataset contains 20 action classes: *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.* Each action class is performed by 10 subjects for 2-3 times. There are 567 videos in total. In [7], they do not use 10 videos because these

Table 1. Performance comparison on MSR action 3D dataset.

Method	Accuracy (%)
Action Graph on Bag of 3D Points [18]	74.70
Histogram of 3D Joints [8]	78.97
Actionlet Ensemble [7]	88.20
Pose-based Recognition [10]	90.22
Moving Pose(MP) [9]	91.70
Maximum Margin Temporal Warping(MMTW) [25]	92.70
Enhanced Sequence Matching (ours)	**94.61**
DL-GSGC [11]	96.70

Table 2. Performance Comparison on UTKinect action dataset.

Method	Accuracy (%)
Skeleton Joint Features [27]	87.90
Histogram of 3D Joints [8]	90.92
Combined features with Random Forests [27]	91.90
Enhanced Sequence Matching (ours)	**93.94**

videos contain highly erroneous positions. For fair comparison we follow the same procedure with [7]. For constructing eMPs, $\tau(\cdot)$ in Equ. (4) is determined in order that the number of candidate eMPs in each action class is around 250 and ρ in Equ. (4) is set to 1.5. As a result of elementary Moving Pose construction, $N_{\mathcal{I}} = 385$ eMPs are constructed on average. For the affine gap function, we set the affine-cost η to 0.5. We use cross-subject test where the videos for half of the subjects are used for training, and the videos of the other half of the subjects for testing.

UTKinect Action Dataset. UTKinect action dataset contains 10 action classes: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands.* Each action class is performed by 10 subjects for 2 times, therefore there are 200 videos in total. This dataset is more challenging than MSR action 3D dataset because each actor in the dataset performs actions in different views. For that reason, we additionally normalize the human pose invariant to viewpoint. Like [8], we define the center of human poses as hip center and rotate each joint position in order that the x-axis and the vector from the left hip center to the right hip center are parallel. For this dataset, $\tau(\cdot)$ is determined in order that the number of candidate eMPs in each action class is around 200 and as a result, $N_{\mathcal{I}} = 385$ eMPs are constructed. ρ is set to 1 and η is set to 0.2. Similar to the MSR action 3D dataset, we use cross-subject testing scheme for evaluating performance.

Table 3. Performance comparison on MSRC-12 iconic gesture dataset.

Method	Accuracy (%)
Nonlinear Markov Models [28]	90.90
Enhanced Sequence Matching (ours)	**96.76**

MSRC-12 Gesture Dataset. MSRC-12 gesture dataset includes 6 iconic and 6 metaphoric gestures performed by 30 people. There are 6,244 gesture instances in 594 videos (719,359 frames in total) in the dataset and instance separation ground-truth is also given. We use 6 iconic gestures (*crouch, put goggle, shoot pistol, throw object, change weapon, kick*) from this dataset, which amounts to 3034 instances. Because the size of this dataset is too big, we sample frames in order that the number of frames in each instance becomes maximally 12 frames. $\tau(\cdot)$ is determined so that the number of candidate eMPs in each action class is around $2,000$ and $N_{\mathcal{I}} = 328$ eMPs are constructed on average. ρ is set to 1 and the affine-cost η is set to 0.5. For performance evaluation, we employ 5-fold leave-person-out cross-validation as in [28]. Specifically, for each fold, instances from 24 subjects are used for training and instances from the remaining 6 subjects are used for testing.

4.2 Comparison with the State-of-the-art

Table 1 shows the action classification accuracies of various algorithms on MSR action 3D dataset. Our Enhanced Sequence Matching (ESM) method achieves the accuracy of 94.61% which is comparable to the state-of-the-art accuracy 96.70% [11] and superior to other methods. Luo et al. [11] concentrate their attention on the class-specific dictionary learning for dealing with intra-class variation rather than the temporal evolution of action. They assume that the action video is localized well therefore they use simply the 3-level temporal pyramid to keep the temporal information of actions. On the other hand, we mainly focus on the temporal evolution of action, therefore, we can handle the weekly localized action (e.g., standing still quite a while at the start of action video or missing a part of an action at the end of video). We expect that the employment of the dictionary learned from [11] in our method instead of the eMP set would improve the performance.

Table 2 and 3 shows the classification results on UTKinect action dataset and MSRC-12 iconic gesture dataset respectively. Our ESM method achieves the state-of-the-art accuracy of 93.94% and 96.76% on both datasets. Especially, in MSRC-12 iconic gesture dataset, we outperform [28] by 6%.

4.3 Discussion

To show the effectiveness of our method in constructing refined action sequence and modeling temporal evolution, we compare our framework with the two variants of DTW.

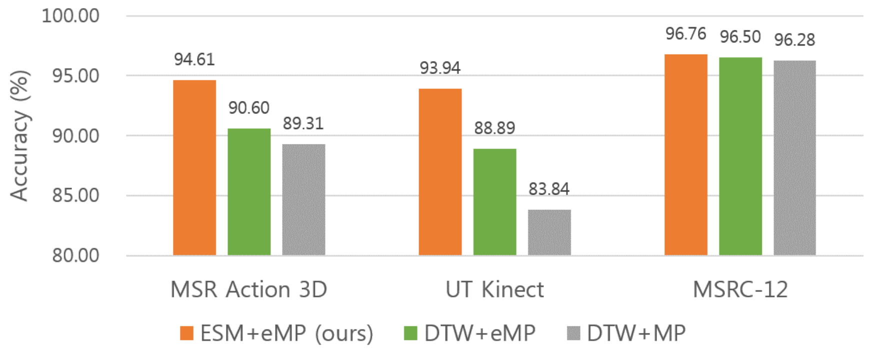

Fig. 5. Classification results of the three datasets using ESM+eMP(our method), DTW+eMP, and DTW+MP.

- DTW+MP: We use the traditional DTW method with the MP descriptor as a frame-level feature. The Euclidean distance between the two MP descriptors is used for matching cost. The result of this framework is the baseline of our experiment.

- DTW+eMP: We use the traditional DTW method with the refined action sequence using eMP. The minus sign of Equ. (17) is used for the matching cost.

- ESM+eMP: This is our framework. We construct the refined action sequence using eMP and classify each sequence by scoring our Enhanced Sequence Matching(ESM) presented in Equ. (16).

Fig. 5 shows the comparing result on the three datasets. Both DTW and ESM model the temporal evolution well. Comparing DTW+MP with DTW+eMP, we can see that refined action sequence is better representation for classification than frame-wise feature. Comparing ESM+eMP with DTW+eMP, the result tells us that ESM is an effective sequence matching method where the noise have potential to deteriorate performance. Example is shown in Fig. 6. Similar action classes such as *high throw* and *hammer* suffer from noise in DTW because the large matching cost would interrupt the classification between these similar actions. However, ESM can ignore the cost from the short-duration noise but penalize the long-duration mismatch so that ESM classifies actions more effectively than DTW in the case of noisy sequence matching.

5 Conclusion

In this paper, we propose a novel framework for action recognition based on 3D skeletal data. Inspired by DNA sequence alignment method used in bioinformatics, we model the new sequence matching method to measure the similarity between two action sequences. We first automatically construct the elementary Moving Pose set by using affinity propagation and then construct refined action sequence which is compact and noise-tempered representation for actions. By

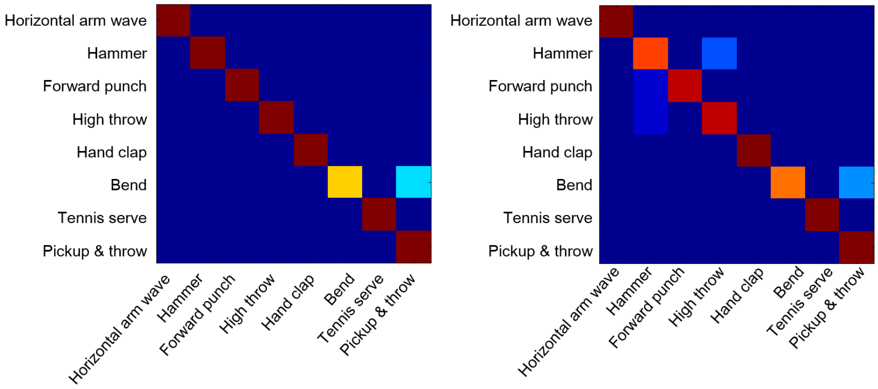

Fig. 6. Confusion matrix of MSR action 3D dataset (AS1). Result of ESM+eMP (*left*) and result of DTW+eMP(*right*).

applying the affine gap function and similarity measure based on both feature and class-distribution to sequence matching score, our method is able to handle noise and action speed variation effectively. Our sequence matching scheme is intuitive and natural and experimental results on three benchmark datasets show that our method works well. We plan to combine part-based recognition approach with our method and to model actions using multiple sequence alignment in the future.

Acknowledgement. This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2011-00166669) and Samsung Electronics Co., Ltd.

References

1. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR, 2011. 3. (2011)
2. Mount, D.W.: Bioinformatics: sequence and genome analysis. Cold Spring Harbor Laboratory Press (2004)
3. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48** (1970) 443-453
4. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of molecular biology **147** (1981) 195-197
5. Vingron, M., Waterman, M.S.: Sequence alignment and penalty choice: Review of concepts, case studies and implications. Journal of Molecular Biology **235** (1994) 1-12
6. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315** (2007)

7. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. (2012) 1290-1297
8. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. (2012) 20-27
9. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: The IEEE International Conference on Computer Vision (ICCV). (2013)
10. Wang, C., Wang, Y., Yuille, A.: An approach to pose-based action recognition. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. (2013) 915-922
11. Luo, J., Wang, W., Qi, H.: Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In: Computer Vision (ICCV), 2013 IEEE International Conference on. (2013) 1809-1816
12. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: in IEEE Conference on Computer Vision and Pattern Recognition(CVPR. (2009)
13. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46** (2002) 131-159
14. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Conditional models for contextual human motion recognition. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. (2005) 1808-1815 Vol. 2
15. Morency, L., Quattoni, A., Darrell, T.: Latent-dynamic discriminative models for continuous gesture recognition. In: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on. (2007) 1-8
16. Lv, F., Nevatia, R.: Recognition and segmentation of 3-d human action using HMM and multi-class adaboost. In Leonardis, A., Bischof, H., Pinz, A., eds.: ECCV 2006. Volume 3954 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 359-372
17. Li, K., Hu, J., Fu, Y.: Modeling complex temporal composition of actionlets for activity prediction. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: Computer Vision ECCV 2012. Volume 7572 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 286-299
18. Li,W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. (2010) 9-14
19. Quattoni, A., Wang, S., Morency, L., Collins, M., Darrell, T.: Hidden conditional random fields. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29** (2007) 1848-1852
20. Han, L., Wu, X., Liang, W., Hou, G., Jia, Y.: Discriminative human action recognition in the learned hierarchical manifold space. Image and Vision Computing **28** (2010) 836-849 Best of Automatic Face and Gesture Recognition 2008
21. Wang, Y., Mori, G.: Hidden part models for human action recognition: Probabilistic versus max margin. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011) 1310-1323
22. Veeraraghavan, A., Roy-chowdhury, A.K.: The function space of an activity. In: in Proc. Comput. Vis. Pattern Recognit. (2006) 959-968
23. Müller, M., Röder, T.: Motion templates for automatic classification and retrieval of motion capture data. In: Proceedings of the 2006 ACM SIGGRAPH Eurographics

Symposium on Computer Animation. SCA '06, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2006) 137-146

24. Yao, B.Z., Zhu, S.C.: Learning deformable action templates from cluttered videos. In: ICCV, IEEE (2009) 1507-1514
25. Wang, J.,Wu, Y.: Learning maximum margin temporal warping for action recognition. In: Computer Vision (ICCV), 2013 IEEE International Conference on. (2013) 2688-2695
26. Fothergill, S., Mentis, H., Kohli, P., Nowozin, S.: Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '12, New York, NY, USA, ACM (2012) 1737-1746
27. Zhu, Y., Chen, W., Guo, G.: Fusing spatiotemporal features and joints for 3d action recognition. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on. (2013) 486-491
28. Lehrmann, A.M., Gehler, P.V., Nowozin, S.: Efficient non-linear markov models for human motion. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, IEEE (2014)