

Accelerated Equivalence Structure Extraction via Pairwise Incremental Search

Seiya Satoh
National Institute of Advanced
Industrial Science and Technology
Koto-ku, Japan
seiya.satoh@aist.go.jp

Yoshinobu Takahashi
University of
Electro-Communications
Chofu, Japan
ytakahashi@ics.lab.uec.ac.jp

Hiroshi Yamakawa
DWANGO Co., Ltd.
Bunkyo-ku, Japan
hiroshi.yamakawa@dwango.co.jp
The Whole Brain Architecture
Initiative

ABSTRACT

Equivalence structure (ES) extraction can allow for finding correspondence relations between different sequential datasets. A K -dimensional ES is a set of K -tuples to specify K -dimensional sequences that are considered equivalent. Whether or not two K -dimensional sequences are equivalent is decided based on comparisons of all of their subsequences. ES extraction can be used for preprocessing for transfer learning or imitation learning, as well as an analysis of multidimensional sequences. A recently proposed method called incremental search (IS) was much faster than brute-force search. However, IS can still take a long time to obtain ESs, because ESs obtained by IS can be subsets of other ESs and such subsets must be removed in the process. In this paper, we propose a new fast method called pairwise incremental search (PIS). In the process of PIS, the aforementioned problem about subsets of ESs does not exist, because the elements of ESs are searched pairwise. As shown by results of two experiments we conducted, PIS was 48 times faster than IS in an experiment using synthetic datasets and 171 times faster in an experiment using motion capture datasets.

CCS CONCEPTS

• **Information systems** → **Data stream mining**; • **Computing methodologies** → *Motif discovery*;

KEYWORDS

equivalence structure; transfer learning; imitation learning

ACM Reference Format:

Seiya Satoh, Yoshinobu Takahashi, and Hiroshi Yamakawa. 2018. Accelerated Equivalence Structure Extraction via Pairwise Incremental Search. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3219819.3220011>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

KDD '18, August 19–23, 2018, London, United Kingdom

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5552-0/18/08.

<https://doi.org/10.1145/3219819.3220011>

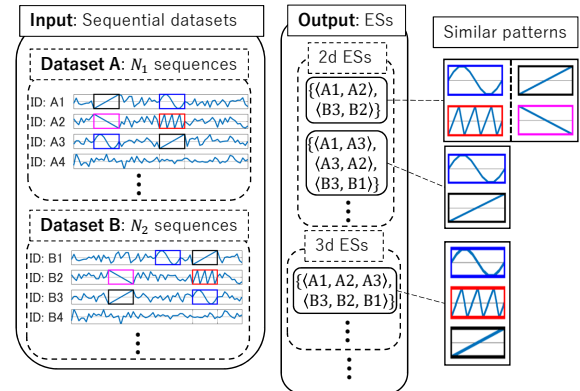


Figure 1: An illustration of ES extraction. The left figure shows the input of the task, and the center figure shows the output. The right figure shows K -dimensional patterns that appear in K -dimensional sequences specified by the tuples of ESs.

1 INTRODUCTION

Equivalence structure (ES) extraction [7, 8] can be useful to find correspondence relations between different sequential datasets, which can be used to analyze sequential datasets. Figure 1 shows an example of ES extraction.

Other applications of ES extraction can be preprocessing for transfer learning [1, 4, 6] or imitation learning [2, 3, 13]. To be more specific, in a task of transfer learning, the experimenter has set the correspondence relation between the variables of different domains, such as a source domain and a target domain. As is the case with imitation learning, the correspondence relation between the variables of a student and those of a teacher has been set by the experimenter in a task of imitation learning. ES extraction may automate the process of finding such correspondence relation.

Motif discovery [5, 11, 12] can be considered similar to ES extraction. Motif discovery techniques allow for the discovery of patterns that appear more than one time in a multidimensional sequence. The techniques also allow for finding a pattern that appears in a multidimensional sequence and a different multidimensional one. However, it must be known how each dimension of one multidimensional sequence corresponds to that of the other different multidimensional sequence. On

the other hand, the purpose of ES extraction is to find such correspondence relations.

The simplest way for ES extraction is brute-force search, but the method is usually not feasible. Therefore, a method called incremental search (IS) was recently proposed, where K -dimensional ESs are searched, making use of $(K - 1)$ -dimensional ESs [7, 8]. However, IS can still take a long time to obtain ESs, because ESs obtained by IS can be subsets of the other ESs, and it is necessary to remove such subsets while performing IS.

In this paper, we propose a new quick method called pairwise incremental search (PIS), where the problem of subsets of ESs does not exist, because the elements of ESs are searched pairwise. We evaluate the performance of PIS using synthetic datasets and motion capture datasets.

This paper is organized as follows. In Section 2, we describe ES extraction itself and the existing methods. In Section 3, we discuss our proposed method. We then evaluate our proposed method using synthetic datasets and motion capture datasets in Section 4. Finally, we offer conclusions and discuss future works in Section 5.

2 EXISTING METHODS FOR ES EXTRACTION

The input of ES extraction is a dataset or datasets of sequences, and the output is a set of multidimensional ESs. A K -dimensional ES is a set of K -tuples whose elements are composed of IDs of the sequences of a dataset to specify K -dimensional sequences. A K -dimensional ES shows which K -dimensional sequences are considered equivalent. As the standard to consider that two K -dimensional sequences are equivalent, we use a dissimilarity function $d(\mathbf{v}_1^{(K)}, \mathbf{v}_2^{(K)})$ used in previous studies [7, 8], where $\mathbf{v}_1^{(K)}$ and $\mathbf{v}_2^{(K)}$ are tuples to specify K -dimensional sequences; all of their subsequences are compared. The details are given in Section 4.1.

ES extraction is rather easy if a sequence and its corresponding sequence have unique patterns. However, this is not always the case, which is one difficulty of ES extraction. For example, in Fig. 1, if we focus on 3-dimensional ES $\{(A1, A2, A2), (B3, B2, B1)\}$, ID A1 corresponds to ID B3. However, there is a sine-wave pattern not only in sequences for IDs A1 and B3 but also in sequences for IDs A3 and B1.

Hereafter, we consider the problem of finding ESs from one dataset. In the case of finding ESs from more than one dataset, a tuple should be composed of IDs of only one dataset.

2.1 Brute-force search

Two methods exist for ES extraction: brute-force search (BFS) and incremental search. BFS is the simplest way to find ESs, where all possible K -dimensional sequences are considered. Given N sequences, the number of all possible K -dimensional sequences to consider is ${}_N P_K$. Therefore, BFS is usually not feasible.

Algorithm 1 Brute-Force Search (BFS)

- 1: **while** a stopping criterion is not satisfied **do**
 - 2: Calculate dissimilarities between all possible K -dimensional sequences
 - 3: Perform clustering to find complete K -dimensional ESs $\mathbf{S}_1^{*(K)}, \dots, \mathbf{S}_{C^*(K)}^{*(K)}$ based on the calculated dissimilarities
 - 4: **end while**
-

2.1.1 Algorithm. Algorithm 1 shows the procedure of BFS. Here, a K -dimensional ES obtained by BFS is called a *complete K -dimensional ES*. A stopping criterion in Step 1 can be a situation where there are no $(K - 1)$ -dimensional ESs, but it is not guaranteed that K -dimensional ESs do not exist if there are no $(K - 1)$ -dimensional ESs.

2.1.2 Clustering techniques for ES extraction. In Step 3 of Algorithm 1, we use the nearest neighbor method, but there are also other clustering techniques, such as the group average method and k -means [10]. However, these methods cannot be used, because a dissimilarity function for ES extraction does not satisfy the triangle inequality.

2.1.3 Property of complete ESs. We call a complete ES $\mathbf{S}_c^{*(K)}$ *permutation-equivalent* to another complete ES $\mathbf{S}_{c'}^{*(K)}$ when the two complete ESs satisfy the following condition:

$$\exists \langle p_1, \dots, p_K \rangle \in \mathbf{P}^{(K)} : \{ \langle s_{p_1}, \dots, s_{p_K} \rangle \mid \langle s_1, \dots, s_K \rangle \in \mathbf{S}_{c'}^{*(K)} \} = \mathbf{S}_c^{*(K)} \quad (1)$$

where $\mathbf{P}^{(K)}$ is the set of all K -permutations of $\{1, \dots, K\}$. For example, if there is a complete ES $\{\langle \#1, \#2, \#3 \rangle, \langle \#4, \#5, \#6 \rangle\}$, then sets $\{\langle \#3, \#2, \#1 \rangle, \langle \#6, \#5, \#4 \rangle\}$ and $\{\langle \#2, \#3, \#1 \rangle, \langle \#5, \#6, \#4 \rangle\}$ are also complete ESs and permutation-equivalent to each other.

2.2 Incremental search

Incremental search (IS) was proposed to address the impracticability of BFS [8]. In the procedure of IS, a K -dimensional ES is obtained from a candidate generated from a $(K - 1)$ -dimensional ES.

2.2.1 Candidate generation. A candidate $\tilde{\mathbf{S}}_{c, \mathbf{c}}^{(K)}$ for a K -dimensional ES is generated by the following function, which uses a $(K - 1)$ -dimensional ES $\tilde{\mathbf{S}}_c^{(K-1)}$ obtained by IS and complete two-dimensional ESs:

$$\begin{aligned} \tilde{\mathbf{S}}_{c, \mathbf{c}}^{(K)} &= g(\tilde{\mathbf{S}}_c^{(K-1)}, \mathbf{c}) \\ &\equiv \left\{ \langle \hat{v}_1, \dots, \hat{v}_{K-1}, i \rangle \mid \langle \hat{v}_1, \dots, \hat{v}_{K-1} \rangle \in \tilde{\mathbf{S}}_c^{(K-1)}, \right. \\ &\quad \left. i \in \left\{ i \mid \bigwedge_{k \in \{1, \dots, K-1\}} \langle \hat{v}_k, i \rangle \in \mathbf{S}_{c_k}^{*(2)} \right\} \right\} \quad (2) \end{aligned}$$

where $\mathbf{c} \in \{\langle c_1, \dots, c_{K-1} \rangle \mid c_1, \dots, c_{K-1} \in \{1, \dots, C^{*(2)}\}\}$ and $C^{*(2)}$ is the number of complete two-dimensional ESs. This function is useful in reducing the number of tuples of a

Algorithm 2 Incremental Search (IS)

```

1:  $K \leftarrow 2$ 
2: Calculate dissimilarities between all possible  $K$ -dimensional sequences
3: Perform clustering to find complete  $K$ -dimensional ESs  $\mathbf{S}_1^{*(K)}, \dots, \mathbf{S}_{C^{*(K)}}^{*(K)}$  based on the dissimilarities
4: Remove ESs that are permutation-equivalent to one of the other ESs
5: Let the remaining ESs be denoted by  $\hat{\mathbf{S}}_1^{(K)}, \dots, \hat{\mathbf{S}}_{\hat{C}^{(K)}}^{(K)}$ 
6: while there exist  $K$ -dimensional ESs do
7:    $K \leftarrow K + 1$ 
8:   for each  $c \in \{1, \dots, \hat{C}^{(K-1)}\}$  do
9:     for each  $\mathbf{c} \in \{\langle c_1, \dots, c_{K-1} \rangle \mid c_1, \dots, c_{K-1} \in \{1, \dots, C^{*(2)}\}\}$  do
10:       $\tilde{\mathbf{S}}_{c, \mathbf{c}}^{(K)} \leftarrow g(\hat{\mathbf{S}}_c^{(K-1)}, \mathbf{c})$ 
11:      Calculate dissimilarities between  $K$ -dimensional sequences specified by  $K$ -tuples of  $\tilde{\mathbf{S}}_{c, \mathbf{c}}^{(K)}$ 
12:      Perform clustering to find  $K$ -dimensional ESs based on the dissimilarities
13:    end for
14:  end for
15:  Remove ESs that are permutation-equivalent to one of the other ESs or subsets of the other ESs
16:  Let the remaining  $K$ -dimensional ESs be denoted by  $\hat{\mathbf{S}}_1^{(K)}, \dots, \hat{\mathbf{S}}_{\hat{C}^{(K)}}^{(K)}$ 
17: end while

```

candidate and has been validated to obtain complete ESs under the following assumption that can be considered to hold to a certain extent[7].

ASSUMPTION 1.

$$\forall k \in \{1, \dots, K\} : \forall c \in \{1, \dots, C^{*(K)}\} : \exists q_{\{k\}} \in \{1, \dots, C^{*(K-1)}\} : \text{drop}(\mathbf{S}_c^{*(K)}, k) \subseteq \mathbf{S}_{q_{\{k\}}}^{*(K-1)} \quad (3)$$

where $\mathbf{S}_1^{*(K)}, \dots, \mathbf{S}_{C^{*(K)}}^{*(K)}$ are all complete K -dimensional ESs, which means that $C^{*(K)}$ is the number of complete K -dimensional ESs, and we define

$$\text{drop}(\mathbf{V}^{(K)}, k) = \{ \langle v_1, \dots, v_{k-1}, v_{k+1}, \dots, v_K \rangle \mid \langle v_1, \dots, v_K \rangle \in \mathbf{V}^{(K)} \}, \quad (4)$$

which drops the k th elements of all the K -tuples of a set $\mathbf{V}^{(K)}$.

2.2.2 Algorithm. Algorithm 2 shows the procedure of IS; Fig. 2 shows an illustration of IS. A clustering technique used in IS is the same as that in BFS (the details are given in Section 2.1.2). We explain the details of Steps 4 and 15 in the next section.

2.3 Properties of ESs obtained by IS

As is the case with complete ESs, an ES obtained by IS can be permutation-equivalent to one of the other ESs. It is necessary to remove such ESs as in Steps 4 and 15 of Algorithm 2. In addition, although it has been proven that ESs obtained by

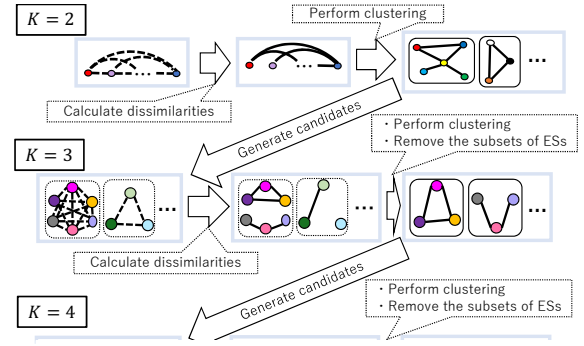


Figure 2: An illustration of IS. A circle, a dashed rounded square, and a solid rounded square represent a tuple, a candidate for a K -dimensional ES, and an ES. A dashed line represents where to calculate a dissimilarity, and a solid line shows that the two tuples are considered equivalent.

IS become complete ESs under Assumption 1, ESs obtained by IS can be subsets of other ESs even if Assumption 1 holds. Therefore, it is necessary to check whether an ES is a subset of one of the other ESs in Step 15 of Algorithm 2. This check must be conducted considering the property of permutation-equivalence. For example, consider one ES $\{\langle \#1, \#2, \#3 \rangle, \langle \#4, \#5, \#6 \rangle, \langle \#7, \#8, \#9 \rangle\}$ and another ES $\{\langle \#6, \#4, \#5 \rangle, \langle \#9, \#7, \#8 \rangle\}$. The latter ES can be seen as a subset of the former ES, because the latter ES is permutation-equivalent to set $\{\langle \#4, \#5, \#6 \rangle, \langle \#7, \#8, \#9 \rangle\}$, which is a subset of the former ES. Therefore, it is necessary to remove the latter ES. The process of the check can cause a slow in processing.

3 PAIRWISE INCREMENTAL SEARCH

Although IS is much faster than BFS, it can still take a long time. One reason is that ESs obtained by IS can be subsets of other ESs, and it is necessary to remove such subsets. In this section, we propose a new fast method called pairwise incremental search (PIS), where the problem of the subsets of ESs does not exist because the elements of ESs are searched pairwise.

First, we define a K -dimensional equivalent pair (EP) as a pair of K -tuples, to specify two K -dimensional sequences that are considered equivalent each other. Figure 3 shows an example of an ES and its EPs. An ES can be constructed from its EPs.

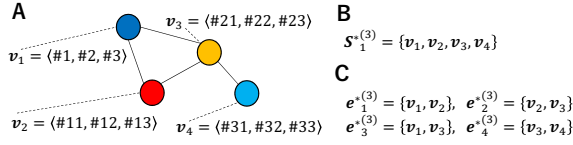


Figure 3: An example of an ES and its EPs: (A) relations between tuples, (B) the ES, and (C) the EPs. A circle represents a tuple, and a solid line between two tuples indicates that the two tuples are equivalent.

3.1 Candidate generation

A candidate $\tilde{e}_{r,r}^{(K)}$ for a K -dimensional EP is generated from a $(K-1)$ -dimensional EP $e_r^{*(K-1)}$ by the following function:

$$\begin{aligned} \tilde{e}_{r,r}^{(K)} &= g'(e_r^{*(K-1)}, r) \\ &\equiv \left\{ \langle v_1^*, \dots, v_{K-1}^*, i \rangle \mid i = \bigwedge_{k \in \{1, \dots, K-1\}} \langle v_k^*, i \rangle \in e_{r_k}^{*(2)}, \right. \\ &\quad \left. \langle v_1^*, \dots, v_{K-1}^* \rangle \in e_r^{*(K-1)} \right\} \end{aligned} \quad (5)$$

where $r \in \{r_1, \dots, r_{K-1} \mid r_1, \dots, r_{K-1} \in \{1, \dots, R^{*(2)}\}\}$, $R^{*(2)}$ is the number of two-dimensional EPs, and r is an ID for a $(K-1)$ -dimensional EP. The biggest difference between the above function and function g in Eq. (2) is that the cardinality of the first argument of g is more than one, while that of g' is always two. Function g' can be derived from Theorem A.2, which we prove in Appendix A, under the following assumption.

ASSUMPTION 2.

$$\begin{aligned} \forall k \in \{1, \dots, K\} : \forall r \in \{1, \dots, R^{*(K)}\} : \\ \exists q_{\{k\}} \in \{1, \dots, R^{*(K-1)}\} : \text{drop}(e_r^{*(K)}, k) = e_{q_{\{k\}}}^{*(K-1)} \end{aligned} \quad (6)$$

where $e_1^{*(K)}, \dots, e_{R^{*(K)}}^{*(K)}$ are all K -dimensional EPs.

For example, if $e_r^{*(K)} = \{\langle \#1, \#2, \#3 \rangle, \langle \#7, \#6, \#4 \rangle\}$, then sets $\{\langle \#2, \#3 \rangle, \langle \#6, \#4 \rangle\}$, $\{\langle \#1, \#3 \rangle, \langle \#7, \#4 \rangle\}$ and $\{\langle \#1, \#2 \rangle, \langle \#7, \#6 \rangle\}$ are all $(K-1)$ -dimensional EPs. As is the case with Assumption 1, how much this assumption holds depends on conditions such as the standard for a dissimilarity function and its parameters, but it is considered that Assumption 2 also holds to a certain extent. Function g' is useful in reducing the number of candidates for EPs, because the simplest way to generate a candidate for a K -dimensional EP under Assumption 2 is to simply add an ID to each of the tuples of a $(K-1)$ -dimensional EP. The number of IDs that can possibly be added to one tuple of a $(K-1)$ -dimensional EP is $N - K + 1$. However, the function g' can allow for the reduction of the number of IDs to be added.

3.2 Algorithm

Algorithm 3 shows the procedure of PIS and Fig. 4 shows an illustration of PIS. If Assumption 2 holds, then all EPs

Algorithm 3 Pairwise Incremental Search (PIS)

```

1:  $K \leftarrow 2$ 
2: Calculate dissimilarities between all possible  $K$ -dimensional sequences
3: Find  $K$ -dimensional equivalent pairs  $e_1^{*(K)}, \dots, e_{R^{*(K)}}^{*(K)}$  from the set of all  $K$ -tuples
4: Remove EPs that are permutation-equivalent to one of the other EPs and let the remaining EPs be denoted by  $e_1^{*(K)}, \dots, e_{\hat{R}^{(K)}}^{*(K)}$ 
5: while there exist  $K$ -dimensional EPs do
6:    $K \leftarrow K + 1$ 
7:   for each  $r \in \{1, \dots, \hat{R}^{(K-1)}\}$  do
8:     for each  $r \in \{r_1, \dots, r_{K-1} \mid r_1, \dots, r_{K-1} \in \{1, \dots, R^{*(2)}\}\}$  do
9:        $\tilde{e}_{r,r}^{(K)} \leftarrow g'(e_r^{*(K-1)}, r)$ 
10:      Calculate a dissimilarity between  $K$ -dimensional sequences specified by  $K$ -tuples of  $\tilde{e}_{r,r}^{(K)}$ 
11:      Check whether  $\tilde{e}_{r,r}^{(K)}$  is a  $K$ -dimensional EP based on the calculated dissimilarity
12:    end for
13:  end for
14:  Remove EPs that are permutation-equivalent to one of the other EPs
15:  Let the remaining  $K$ -dimensional EPs be denoted by  $e_1^{*(K)}, \dots, e_{\hat{R}^{(K)}}^{*(K)}$ 
16: end while

```

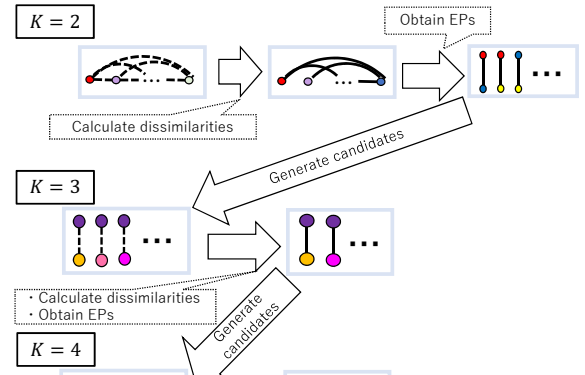


Figure 4: An illustration of PIS. A circle and a dashed line represent a tuple and where to calculate a dissimilarity, respectively. A solid line shows that the two tuples are considered equivalent.

are obtained. In experiments in this paper, we construct only the highest-dimensional ESs from EPs after performing PIS. We explain our reasons for this in the next section.

3.3 Redundancy of ESs

If a K -dimensional ES $\widehat{\mathbf{S}}_c^{(K)}$ is obtained by IS, whose elements are $\langle v_{1,1}, \dots, v_{1,K} \rangle, \dots, \langle v_{M,1}, \dots, v_{M,K} \rangle$, then

$$\begin{aligned} \forall k \in \{2, \dots, K-1\} : \forall \{c_1, \dots, c_k\} \in \mathbf{C}^{(k)} : \\ \exists c' \in \{1, \dots, \widehat{\mathbf{C}}^{(k)}\} : \forall m \in \{1, \dots, M\} : \\ \langle v_{m,c_1}, \dots, v_{m,c_k} \rangle \in \widehat{\mathbf{S}}_{c'}^{(k)} \end{aligned} \quad (7)$$

where $\widehat{\mathbf{S}}_c^{(k)}$ is a k -dimensional ES, $\widehat{\mathbf{C}}^{(k)}$ is the number of k -dimensional ESs obtained by IS, and $\mathbf{C}^{(k)}$ is the set of all k -combinations of $\{1, \dots, K\}$. The number of possible tuples for $\langle v_{m,c_1}, \dots, v_{m,c_k} \rangle$ in the above equation is $M \times \sum_{k=2}^{K-1} K C_k$. Such tuples are redundant, but the tuples of the highest-dimensional ESs can never be redundant and usually have the most important information. From this perspective, we construct only the highest-dimensional ESs from the highest-dimensional EPs after performing PIS. It is also considered that in some cases, having EPs without constructing ESs is enough.

3.4 Advantages of PIS compared to IS

Compared to IS, these are the following advantages of PIS:

- PIS does not require a clustering technique.
- PIS does not have the problem of subsets of ESs as discussed in Section 2.3.

Both advantages stem from a feature of PIS that the cardinality of an EP is always two.

4 EXPERIMENTS

We conducted two experiments to evaluate our proposed method, PIS, comparing it with IS. We used synthetic datasets for the first experiment and motion capture datasets (obtained from mocap.cs.cmu.edu) for the second experiment.

Both experiments address the problem of finding correspondence relations between two datasets. Therefore, we considered tuples composed of IDs of only one dataset. In addition, we compared a multidimensional sequence from one dataset with that from the other dataset, but not with a sequence from the same dataset. In this case, the number of comparisons for the simplest BFS is $N_1 P_K \times N_2 P_K$ where N_1 and N_2 are the number of sequences for each dataset, respectively. When we consider the following property of a dissimilarity function:

$$\begin{aligned} \forall \langle p_1, \dots, p_K \rangle \in \mathbf{P}^{(K)} : \\ d(\mathbf{v}_1, \mathbf{v}_2) = d(\langle v_{1,p_1}, \dots, v_{1,p_K} \rangle, \langle v_{2,p_1}, \dots, v_{2,p_K} \rangle), \end{aligned} \quad (8)$$

the number $n_{BFS}^{(K)}$ of comparisons for BFS is

$$n_{BFS}^{(K)} = \frac{N_1 P_K \times N_2 P_K}{K!}. \quad (9)$$

In the procedure of IS and PIS, the calculations of the dissimilarity function $d(\mathbf{v}_1, \mathbf{v}_2)$ are independent, so we used the parfor function in the Parallel Computing Toolbox version 6.7 in the MATLAB R2015b to compute the dissimilarities in parallel. In addition, clustering in Step 12 of IS can also be computed in parallel, so we used the parfor function in this

process as well. We used a computer whose CPU is Intel(R) Xeon(R) CPU E5-2600 v3 @ 2.60GHz. In the next section, we describe the details of the dissimilarity function we used.

4.1 Dissimilarity function

As the standard to determine whether K -dimensional sequences are equivalent, we used a dissimilarity function that uses mean-square values (MSVs) of Euclidean distances between all subsequences. Here, we consider N sequences $\{\mathbf{x}_i = (x_i^{(1)}, \dots, x_i^{(T)})\}_{i \in \{1, \dots, N\}}$ whose lengths are T , and we define the following subsequence of a sequence \mathbf{x}_k

$$\begin{aligned} \mathbf{z}_k^{(t)} &\equiv (z_k^{(t,1)}, \dots, z_k^{(t,\tau)})^{\text{tr}} \\ &= (x_k^{(t)}, \dots, x_k^{(t+\tau-1)})^{\text{tr}} - \frac{1}{\tau} \sum_{t'=1}^{\tau} x_k^{(t+t'-1)} \end{aligned} \quad (10)$$

where τ is the length of a subsequence, t is the starting point, k is an ID for a sequence, and \mathbf{a}^{tr} is the transportation of a vector \mathbf{a} . The second term ensures that the mean of $\mathbf{z}_k^{(t)}$ is 0.

The dissimilarity $d(\mathbf{v}_1, \mathbf{v}_2)$ between two K -dimensional sequences specified by two K -tuples \mathbf{v}_1 and \mathbf{v}_2 is calculated as:

$$d(\mathbf{v}_1, \mathbf{v}_2) = 1 - \frac{1}{\beta} \sum_{t=1}^{T-\tau+1} w_{\mathbf{v}_1}^{(t)} h_{\mathbf{v}_1, \mathbf{v}_2}^{(t)} + w_{\mathbf{v}_2}^{(t)} h_{\mathbf{v}_2, \mathbf{v}_1}^{(t)} \quad (11)$$

where

$$\beta \equiv \sum_{t=1}^{T-\tau+1} w_{\mathbf{v}_1}^{(t)} + w_{\mathbf{v}_2}^{(t)}, \quad (12)$$

$$w_{\mathbf{v}_1}^{(t)} \equiv \frac{1}{\tau} \sum_{t'=1}^{\tau} \sqrt{\sum_{k=1}^K \{z_{v_1, k}^{(t, t')}\}^2}, \quad (13)$$

$$h_{\mathbf{v}_1, \mathbf{v}_2}^{(t)} \equiv h(\theta_{MSV} - MSV_{\mathbf{v}_1, \mathbf{v}_2}^{(t)}), \quad (14)$$

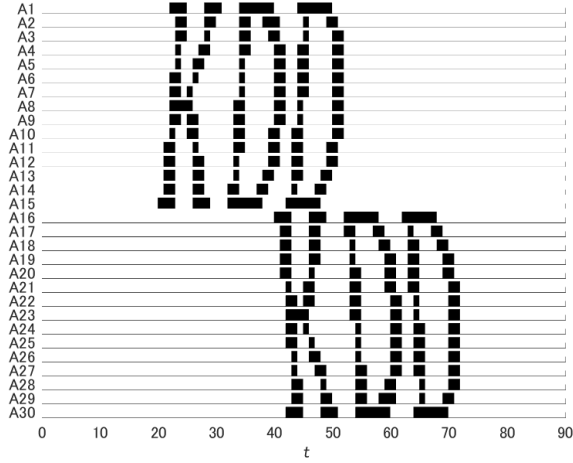
$$MSV_{\mathbf{v}_1, \mathbf{v}_2}^{(t)} \equiv \min(MSV_{\mathbf{v}_1, \mathbf{v}_2}^{(t, t')} | t' = 1, \dots, T - \tau + 1), \quad (15)$$

$$MSV_{\mathbf{v}_1, \mathbf{v}_2}^{(t, t')} \equiv \frac{1}{\tau K} \sum_{k=1}^K |z_{v_1, k}^{(t)} - z_{v_2, k}^{(t')}|^2, \quad (16)$$

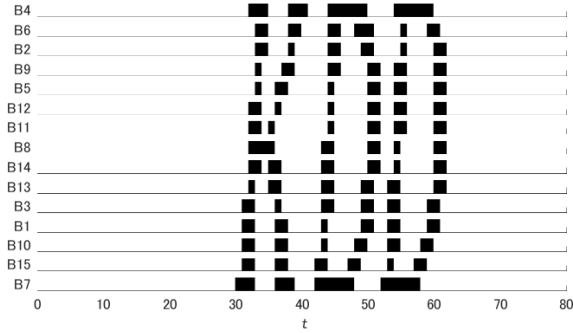
$\mathbf{v}_1 \equiv \langle v_{1,1}, \dots, v_{1,K} \rangle$, $\mathbf{v}_2 \equiv \langle v_{2,1}, \dots, v_{2,K} \rangle$, and h is the Heaviside step function. The value $d(\mathbf{v}_1, \mathbf{v}_2)$ is less than or equal to 1 because of β . $h_{\mathbf{v}_1, \mathbf{v}_2}^{(t)}$ is 1 if $MSV_{\mathbf{v}_1, \mathbf{v}_2}^{(t)} < \theta_{MSV}$ and 0 otherwise. $w_{\mathbf{v}_1}^{(t)}$ becomes larger as the variations in the K -dimensional subsequence $(z_{v_1, 1}^{(t)}, \dots, z_{v_1, K}^{(t)})$ becomes larger. Therefore, if $h_{\mathbf{v}_1, \mathbf{v}_2}^{(t)}$ is 1 where $w_{\mathbf{v}_1}^{(t)}$ is large, then it is likely to be determined that the K -dimensional sequence specified by \mathbf{v}_1 and that by \mathbf{v}_2 are equivalent. Two K -dimensional sequences specified by \mathbf{v}_1 and \mathbf{v}_2 are determined equivalent if $d(\mathbf{v}_1, \mathbf{v}_2)$ is less than a threshold θ_E . We used different values for the two thresholds θ_{MSV} and θ_E for each experiment.

4.2 Experiment 1

For the first experiment, we used two synthetic datasets (synthetic datasets 1 and 2), shown in Fig. 5. IDs for synthetic dataset 1 are A1, ..., A30, while those for synthetic dataset 2



(a) Synthetic dataset 1



(b) Synthetic dataset 2

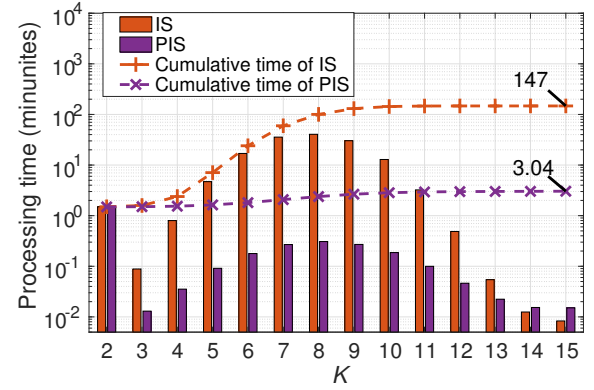
Figure 5: Synthetic datasets

are B1, ..., B15. The length of the sequences whose IDs are A1, ..., A30 is 90, while that of the sequences whose IDs are B1, ..., B15 is 80. There is a pattern from $t = 20$ to $t = 52$ in sequences whose IDs are A1, ..., A15, and another pattern from $t = 40$ to $t = 72$ in sequences whose IDs are A16, ..., A30. Note that the latter pattern is the same as the former pattern in reverse order. The same pattern also appears from $t = 30$ to $t = 62$ in synthetic dataset 2, but the order of the sequences is different.

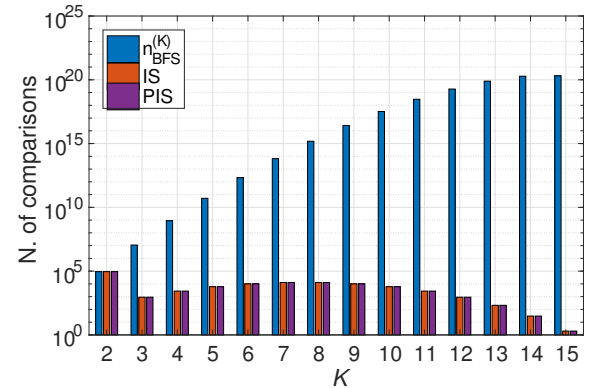
In Fig. 5, black points mean that the values are 1, while white points mean that the values are 0. We added uniformly distributed random numbers in the interval $(-0.01, 0.01)$ to the sequences.

For preprocessing, we normalized all sequences so that the means and standard deviations would be 0 and 1, respectively. We set the length of the subsequences to 20, so we had 71 subsequences for each of the sequences of synthetic dataset 1 and 61 subsequences for each of the sequences of synthetic dataset 2. We used the dissimilarity function discussed in Section 4.1. We set both thresholds θ_{MSV} and θ_E to 0.01.

Figure 6a shows the processing times for IS and PIS. The total processing time of IS was 147 min, while that of PIS



(a) Processing times



(b) Numbers of comparisons

Figure 6: Results for synthetic datasets

was 3.04 min, which means that PIS was 48 times faster than IS.

Figure 6b shows $n_{BFS}^{(K)}$ in Eq. (9), the numbers of comparisons for IS and those for PIS. We see that BFS was not feasible, and the numbers of comparisons for IS were exactly the same as those for PIS.

IS allowed us to obtain one 15-dimensional ES (shown in Table 1), which was the highest-dimensional. PIS allowed us to obtain two 15-dimensional EPs (shown in Table 2), which were the highest-dimensional. Using the EPs, we could construct the same ES in Table 1 spending less than a second.

4.3 Experiment 2

For the second experiment, we used two motion capture datasets that were created from two walking persons. The two datasets were the same datasets used in previous studies [7, 8]. We obtained the datasets with file names 07_01.c3d and 07_02.c3d from mocap.cs.cmu.edu. Using these datasets, which here we call mocap datasets 1 and 2, we examined whether ES extraction could allow us to find correspondence relations.

For preprocessing, we used the following preprocessing, which was the same method used in a previous study [7]. (See

Table 1: Highest-dimensional ES obtained by IS for synthetic datasets

15-dimensional ES 1	{(A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15), (A30, A29, A28, A27, A26, A25, A24, A23, A22, A21, A20, A19, A18, A17, A16), (B4, B6, B2, B9, B5, B12, B11, B8, B14, B13, B3, B1, B10, B15, B7)}
---------------------	--

Table 2: Highest-dimensional EPs obtained by PIS for synthetic datasets

15-dimensional EP 1	{(A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15), (B4, B6, B2, B9, B5, B12, B11, B8, B14, B13, B3, B1, B10, B15, B7)}
15-dimensional EP 2	{(A16, A17, A18, A19, A20, A21, A22, A23, A24, A25, A26, A27, A28, A29, A30), (B7, B15, B10, B1, B3, B13, B14, B8, B11, B12, B5, B9, B2, B6, B4)}

the details in the study.) We first normalized all sequences so that the means and standard deviations would be 0 and 1, respectively. Next, we used the simple moving average technique as a low-pass filter, which uses the mean of continuous n points of each sequence. Then, we down-sampled each sequence by n . The number n was set to six considering errors between original sequences and sequences after the preprocessing. Finally, to reduce the number of sequences, we merged two sequences of a dataset if the two sequences satisfy the following condition

$$0.2 > \frac{1}{T} \sum_{j=1}^3 (\mathbf{r}_j^{(i_1)} - \mathbf{r}_j^{(i_2)})^T (\mathbf{r}_j^{(i_1)} - \mathbf{r}_j^{(i_2)}) \quad (17)$$

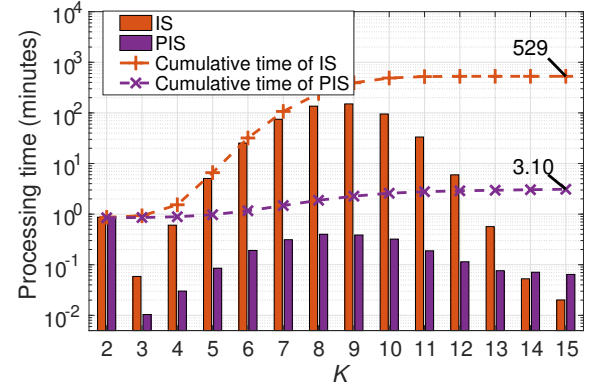
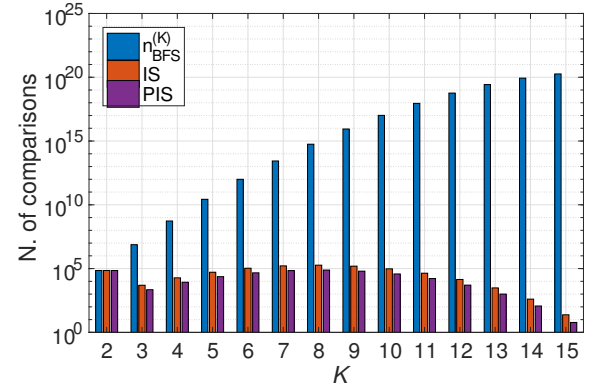
where i_1 and i_2 are the IDs of two sequences and T is the length of sequences; we refer to the two sequences as T -by-3 matrices $\mathbf{R}^{(i_1)} = (\mathbf{r}_1^{(i_1)}, \mathbf{r}_2^{(i_1)}, \mathbf{r}_3^{(i_1)})$, $\mathbf{R}^{(i_2)} = (\mathbf{r}_1^{(i_2)}, \mathbf{r}_2^{(i_2)}, \mathbf{r}_3^{(i_2)})$, since each of the sequences has 3-dimensional information. After the preprocessing, we obtained 22 sequences for mocap dataset 1 and 18 sequences for mocap dataset 2. We assigned IDs A1, ..., A22 to the sequences for mocap dataset 1, and IDs B1, ..., B18 to the sequences for mocap dataset 2.

We set the length of the subsequences to 10. We used the same dissimilarity function used in Experiment 1. We set the thresholds θ_{MSV} and θ_E to 0.06 and 0.5, respectively.

Figure 7a shows the processing times for IS and PIS. The total processing time of IS was 529 min, while that of PIS was 3.10 min, which means that PIS was 171 times faster than IS.

Figure 7b shows $n_{BFS}^{(K)}$ in Eq. (9), the numbers of comparisons for IS and those for PIS. We see that BFS was not feasible in this experiment, as was the case in Experiment 1. The numbers of comparisons for IS were slightly larger than those for PIS when $K > 2$. It can be considered that the difference came from the difference between Assumptions 1 and 2 and the difference between the noise level of the synthetic datasets and that of the mocap datasets.

IS allowed us to obtain three 15-dimensional ESs (15-dimensional ESs 1, 2, and 3) shown in Table 3, which were the highest-dimensional. PIS allowed us to obtain six 15-dimensional EPs, which were the highest-dimensional; that is, the number of the dimensions of the EPs was the same

**(a) Processing times****(b) Numbers of comparisons****Figure 7: Results for mocap datasets**

as that of ESs obtained by IS. We could construct three 15-dimensional ESs (15-dimensional ESs 4, 5, and 6), shown in Table 4, from the six EPs, spending less than a second. Here, we call the 15-dimensional ESs 1, ..., 6. We can see that ESs 4, 5, and 6 are subsets of ESs 1, 2, and 3, respectively. Therefore, ESs obtained by PIS were not exactly the same ESs obtained by IS, but they were similar. The differences between the ESs obtained by PIS and those by IS can be summarized as follows:

Table 3: Highest-dimensional ESs obtained by IS for mocap datasets

15-dimensional ES 1	{(A1, A6, A7, A9, A10, A12 , A14, A15 , A16, A17 , A18, A19, A20, A21, A22), (A1, A6, A7, A9, A10, A12 , A14, A17 , A16, A15 , A18, A19, A20, A21, A22), (A1, A6, A7, A9, A12, A10 , A14, A15 , A16, A17 , A18, A19, A20, A21, A22), (A1, A6, A7, A9, A12, A10 , A14, A17 , A16, A15 , A18, A19, A20, A21, A22), (B4, B6, B9, B11, B5, B16, B12, B14, B10, B15, B1, B17, B2, B18, B8), (B4, B6, B9, B11, B16, B5, B12, B14, B10, B15, B1, B17, B2, B18, B8), (B4, B6, B9, B11, B16, B5, B12, B15, B10, B14, B1, B17, B2, B18, B8)}
15-dimensional ES 2	{(A4, A6, A7 , A8, A9, A10, A12, A15 , A16, A17 , A18, A19, A20, A21, A22), (A4, A6, A7 , A8, A9, A10, A12, A17 , A16, A15 , A18, A19, A20, A21, A22), (A4, A6, A7 , A8, A9, A12, A10, A15 , A16, A17 , A18, A19, A20, A21, A22), (A4, A6, A7 , A8, A9, A12, A10, A17 , A16, A15 , A18, A19, A20, A21, A22), (A4, A6, A11 , A8, A9, A10, A12, A15 , A16, A17 , A18, A19, A20, A21, A22), (A4, A6, A11 , A8, A9, A10, A12, A17 , A16, A15 , A18, A19, A20, A21, A22), (A4, A6, A11 , A8, A9, A12, A10, A15 , A16, A17 , A18, A19, A20, A21, A22), (A4, A6, A11 , A8, A9, A12, A10, A17 , A16, A15 , A18, A19, A20, A21, A22), (B4, B6, B9, B3, B11, B5, B16, B14, B10, B15, B1, B17, B2, B18, B8), (B4, B6, B9, B3, B11, B5, B16, B14, B10, B15, B1, B17, B7, B18, B8), (B4, B6, B9, B3, B11, B5, B16, B15, B10, B14, B1, B17, B2, B18, B8), (B4, B6, B9, B3, B11, B5, B16, B15, B10, B14, B1, B17, B7, B18, B8), (B4, B6, B9, B3, B11, B16, B5, B14, B10, B15, B1, B17, B2, B18, B8), (B4, B6, B9, B3, B11, B16, B5, B14, B10, B15, B1, B17, B7, B18, B8), (B4, B6, B9, B3, B11, B16, B5, B15, B10, B14, B1, B17, B2, B18, B8), (B4, B6, B9, B3, B11, B16, B5, B15, B10, B14, B1, B17, B7, B18, B8)}
15-dimensional ES 3	{(A6, A7, A8, A9, A10, A12 , A14, A15 , A16, A17 , A18, A19, A20, A21, A22), (A6, A7, A8, A9, A10, A12 , A14, A17 , A16, A15 , A18, A19, A20, A21, A22), (A6, A7, A8, A9, A12, A10 , A14, A15 , A16, A17 , A18, A19, A20, A21, A22), (A6, A7, A8, A9, A12, A10 , A14, A17 , A16, A15 , A18, A19, A20, A21, A22), (B6, B9, B3, B11, B5, B16, B12, B14, B10, B15, B1, B17, B2, B18, B8), (B6, B9, B3, B11, B5, B16, B12, B15, B10, B14, B1, B17, B2, B18, B8), (B6, B9, B3, B11, B16, B5, B12, B14, B10, B15, B1, B17, B2, B18, B8), (B6, B9, B3, B11, B16, B5, B12, B15, B10, B14, B1, B17, B2, B18, B8)}

Table 4: Highest-dimensional ESs obtained by PIS for mocap datasets

15-dimensional ES 4	{(A1, A6, A7, A9, A10, A12 , A14, A15 , A16, A17 , A18, A19, A20, A21, A22), (B4, B6, B9, B11, B5, B16, B12, B14, B10, B15, B1, B17, B2, B18, B8)}
15-dimensional ES 5	{(A4, A6, A7, A8, A9, A10, A12, A15 , A16, A17 , A18, A19, A20, A21, A22), (A4, A6, A7, A8, A9, A10, A12, A17 , A16, A15 , A18, A19, A20, A21, A22), (A4, A6, A11, A8, A9, A10, A12, A15 , A16, A17 , A18, A19, A20, A21, A22), (A4, A6, A11, A8, A9, A10, A12, A17 , A16, A15 , A18, A19, A20, A21, A22), (B4, B6, B9, B3, B11, B5, B16, B14, B10, B15, B1, B17, B2, B18, B8), (B4, B6, B9, B3, B11, B5, B16, B15, B10, B14, B1, B17, B2, B18, B8)}
15-dimensional ES 6	{(A6, A7, A8, A9, A10, A12 , A14, A15 , A16, A17 , A18, A19, A20, A21, A22), (B6, B9, B3, B11, B5, B16, B12, B14, B10, B15, B1, B17, B2, B18, B8)}

- ESs 4 and 6 imply that none of the IDs can be switched, while ESs 1 and 3 imply that IDs A10 and A15 are switchable to IDs A12 and A17, respectively.
- ES 5 implies that IDs A7 and A15 are switchable to IDs A11 and A17, respectively, while ES 1 implies that ID A10 is also switchable to ID A12.

Note that ESs obtained by IS may not be complete ESs, as is the case with those by PIS. In addition, because we have not had a measure of the usefulness of ESs, it is unclear that ESs should be complete, as discussed in a previous study [7].

Figure 8 shows ES 5 drawn in the original 3-dimensional space. Since almost all of the red lines in the figure are parallel to the x -axis, it is concluded that PIS allowed us to find a quality correspondence relation between the two mocap datasets. Note that we did not use positional relations, since all of the sequences were normalized.

5 CONCLUSION

In this paper, we proposed a new method called pairwise incremental search (PIS), where the elements of ESs are

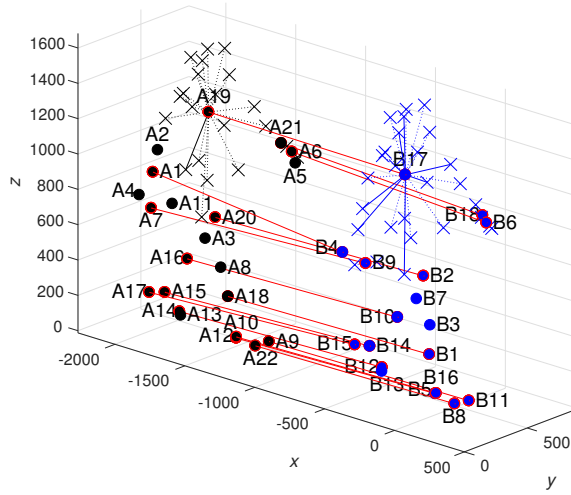


Figure 8: 15-dimensional ES 5 drawn in the original three-dimensional space. Black points represent sequences for IDs A1, ..., A22 at $t = 1$, while blue points represent sequences for IDs B1, ..., B18 at $t = 150$. Cross marks represent sequences that were merged in the preprocessing. The IDs at each position in the tuples of the ES are connected with red lines.

searched pairwise. There had been two existing methods, brute-force search and incremental search (IS), with the latter usually being much faster than brute-force search. In the first experiment using synthetic datasets, the total processing time for PIS was 48 times faster than that for IS, and PIS allowed us to obtain the same ESs as those obtained by IS. In the second experiment using two motion capture datasets, the total processing time for PIS was 171 times faster than that for IS, and PIS allowed us to obtain ESs similar to those obtained by IS. We consider that ESs obtained by PIS implied quality correspondence relations between the two motion capture datasets.

In the future, we plan to examine the influences of parameters to find optimum parameters, since we conducted each experiment with only one setting for parameters. However, to do this, a measure of the usefulness of an ES will be necessary, and there has not yet been such a measure. In addition, we plan to propose a faster method, because the number of calculations of dissimilarities for PIS was almost the same as that for IS, and it can also cause a slow process.

ACKNOWLEDGMENTS

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). The motion capture data used in this paper was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

A VALIDATION OF CANDIDATE GENERATION FOR PIS

To begin with, we define $[i]$ to be $\{1, \dots, i\}$. Under Assumption 2, we can derive the following lemma.

LEMMA A.1. *Under Assumption 2,*

$$\begin{aligned} \forall r \in [R^{*(K)}] : \forall i_1 \in [K] : \forall i_2 \in [K] \setminus i_1 : \\ \exists q_{\{i_1, i_2\}} \in [R^{*(2)}] : \\ \text{drop}(e_r^{*(K)}, [K] \setminus \{i_1, i_2\}) = e_{q_{\{i_1, i_2\}}}^{*(2)} \end{aligned} \quad (18)$$

where $e_1^{*(K)}, \dots, e_{R^{*(K)}}^{*(K)}$ are all K -dimensional EPs.

For example, if there is a four-dimensional EP $e_1^{*(4)} = \{\langle \#1, \#2, \#3, \#4 \rangle, \langle \#5, \#6, \#7, \#8 \rangle\}$, then $\text{drop}(e_1^{*(4)}, [4] \setminus \{1, 2\})$, $\text{drop}(e_1^{*(4)}, [4] \setminus \{1, 3\})$ and $\text{drop}(e_1^{*(4)}, [4] \setminus \{3, 4\})$ are sets $\{\langle \#1, \#2 \rangle, \langle \#5, \#6 \rangle\}$, $\{\langle \#1, \#3 \rangle, \langle \#5, \#7 \rangle\}$ and $\{\langle \#3, \#4 \rangle, \langle \#7, \#8 \rangle\}$, respectively, and they are all two-dimensional EPs.

PROOF. We use mathematical induction [9] to prove Lemma A.1. Under Assumption 2, obviously,

$$\begin{aligned} \forall r \in [R^{*(K)}] : \exists q_{\{p_K\}} \in [R^{*(K-1)}] : \\ e_{q_{\{p_K\}}}^{*(K-1)} = \text{drop}(e_r^{*(K)}, p_K) \end{aligned} \quad (19)$$

where $\langle p_1, \dots, p_K \rangle$ is a K -permutation of $[K]$ and $3 < K$. Consider a particular K -dimensional EP $e_r^{*(K)}$ and a particular $(K-1)$ -dimensional EP $e_{q_{\{p_K\}}}^{*(K-1)}$ that satisfies $\text{drop}(e_r^{*(K)}, p_K) = e_{q_{\{p_K\}}}^{*(K-1)}$. If we additionally drop the p_{K-1} th elements of the two tuples of $e_{q_{\{p_K\}}}^{*(K-1)}$, we have

$$\begin{aligned} \exists q_{\{p_{K-1}, p_K\}} \in [R^{*(K-2)}] : \\ e_{q_{\{p_{K-1}, p_K\}}}^{*(K-2)} = \text{drop}(e_{q_{\{p_K\}}}^{*(K-1)}, p_{K-1}). \end{aligned} \quad (20)$$

From Eq. (19), the right-hand side can be rewritten and we have

$$\begin{aligned} \exists q_{\{p_{K-1}, p_K\}} \in [R^{*(K-2)}] : \\ e_{q_{\{p_{K-1}, p_K\}}}^{*(K-2)} &= \text{drop}(\text{drop}(e_r^{*(K)}, p_K), p_{K-1}) \\ &= \text{drop}(e_r^{*(K)}, \{p_{K-1}, p_K\}). \end{aligned} \quad (21)$$

Next, we assume:

$$\begin{aligned} \forall r \in [R^{*(K)}] : \exists q_{\{p_{K-k}, \dots, p_K\}} \in [R^{*(K-k-1)}] : \\ \text{drop}(e_r^{*(K)}, \{p_{K-k}, \dots, p_K\}) = e_{q_{\{p_{K-k}, \dots, p_K\}}}^{*(K-k-1)} \end{aligned} \quad (22)$$

where $k \in [K-4]$. Consider a particular K -dimensional EP $e_r^{*(K)}$ and a particular $(K-k-1)$ -dimensional EP $e_{q_{\{p_{K-k}, \dots, p_K\}}}^{*(K-k-1)}$ that satisfy $\text{drop}(e_r^{*(K)}, \{p_{K-k}, \dots, p_K\}) = e_{q_{\{p_{K-k}, \dots, p_K\}}}^{*(K-k-1)}$. If we also drop the p_{K-k-1} th elements of all the tuples of $e_{q_{\{p_{K-k}, \dots, p_K\}}}^{*(K-k-1)}$, we have

$$\begin{aligned} \exists q_{\{p_{K-k-1}, \dots, p_K\}} \in [R^{*(K-k-2)}] : \\ e_{q_{\{p_{K-k-1}, \dots, p_K\}}}^{*(K-k-2)} = \text{drop}(e_{q_{\{p_{K-k}, \dots, p_K\}}}^{*(K-k-1)}, p_{K-k-1}). \end{aligned} \quad (23)$$

From Eq. (22), the right-hand side can be rewritten and we have

$$\begin{aligned} \exists q_{\{p_{K-k-1}, \dots, p_K\}} &\in [R^{*(K-k-2)}] : \\ e_{q_{\{p_{K-k-1}, \dots, p_K\}}}^{*(K-k-2)} &= \text{drop}(\text{drop}(e_r^{*(K)}, \{p_{K-k}, \dots, p_K\}), \\ &\quad p_{K-k-1}) \\ &= \text{drop}(e_r^{*(K)}, \{p_{K-k-1}, \dots, p_K\}). \end{aligned} \quad (24)$$

From Eqs. (21) and (24), we obtain:

$$\begin{aligned} \forall k \in [K-3] : \forall r \in [R^{*(K)}] : \\ \exists q_{\{p_{K-k}, \dots, p_K\}} &\in [R^{*(K-k-1)}] : \\ \text{drop}(e_r^{*(K)}, \{p_{K-k}, \dots, p_K\}) &= e_{q_{\{p_{K-k}, \dots, p_K\}}}^{*(K-k-1)}. \end{aligned} \quad (25)$$

Hence, we see that

$$\begin{aligned} \forall r \in [R^{*(K)}] : \exists q_{\{p_{K-k}, \dots, p_K\}} &\in [R^{*(K-k-1)}] : \\ \text{drop}(e_r^{*(K)}, \{p_3, \dots, p_K\}) &= e_{q_{\{p_3, \dots, p_K\}}}^{*(2)}. \end{aligned} \quad (26)$$

□

Lemma A.1 tells us that the sets of tuples composed of the i th and j th IDs of the tuples of a K -dimensional EP are all two-dimensional EPs. Considering Lemma A.1, we can derive the following theorem.

THEOREM A.2. *Under Assumption 2,*

$$\begin{aligned} \exists r_1 \in [R^{*(2)}] : \dots \exists r_{K-1} \in [R^{*(2)}] : \\ e_r^{*(K)} = \tilde{e}_{q_K, \mathbf{r}^{(K-1)}}^{(K)} \end{aligned} \quad (27)$$

where

$$\begin{aligned} \tilde{e}_{q_K, \mathbf{r}^{(K-1)}}^{(K)} \equiv \left\{ \left\langle v_1, \dots, v_{K-1}, i \right\rangle \left| \bigwedge_{k \in [K-1]} \langle v_k, i \rangle \in e_{r_k}^{*(2)}, \right. \right. \\ \left. \left. \langle v_1, \dots, v_{K-1} \rangle \in e_{q_K}^{*(K-1)} \right\}, \end{aligned} \quad (28)$$

$$\text{drop}(e_r^{*(K)}, K) = e_{q_K}^{*(K-1)}, \quad (29)$$

$$\mathbf{r}^{(K-1)} \equiv \langle r_1, \dots, r_{K-1} \rangle, \quad (30)$$

and $e_r^{*(K)}$ and $e_{q_K}^{*(K-1)}$ are a particular K -dimensional EP and a particular $(K-1)$ -dimensional EP that satisfy Eq. (29), respectively.

PROOF. From Eq. (29), it is clear that

$$\begin{aligned} \forall \langle v_1^*, \dots, v_K^* \rangle \in e_r^{*(K)} : \exists \langle v_1, \dots, v_{K-1} \rangle \in e_{q_K}^{*(K-1)} : \\ \bigwedge_{k \in [K-1]} v_k = v_k^*. \end{aligned} \quad (31)$$

From Lemma A.1, we see that

$$\begin{aligned} \forall k \in [K-1] : \exists r_k \in [R^{*(2)}] : \forall \langle v_1^*, \dots, v_K^* \rangle \in e_r^{*(K)} : \\ \text{drop}(\langle v_1^*, \dots, v_K^* \rangle, [K] \setminus \{k, K\}) \in e_{r_k}^{*(2)}. \end{aligned} \quad (32)$$

Let r_k^* be a r_k that satisfies

$$\begin{aligned} \forall \langle v_1^*, \dots, v_K^* \rangle \in e_r^{*(K)} : \\ \text{drop}(\langle v_1^*, \dots, v_K^* \rangle, [K] \setminus \{k, K\}) \in e_{r_k^*}^{*(2)}. \end{aligned} \quad (33)$$

We then see that

$$\forall k \in [K-1] : \forall \langle v_1^*, \dots, v_K^* \rangle \in e_r^{*(K)} : \langle v_k^*, v_K^* \rangle \in e_{r_k^*}^{*(2)}. \quad (34)$$

From this equation, we find that

$$\forall \langle v_1^*, \dots, v_K^* \rangle \in e_r^{*(K)} : \bigwedge_{k \in [K-1]} \langle v_k^*, v_K^* \rangle \in e_{r_k^*}^{*(2)}. \quad (35)$$

Therefore, we have

$$\begin{aligned} \exists r_1 \in [R^{*(2)}] : \dots \exists r_{K-1} \in [R^{*(2)}] : \\ e_r^{*(K)} = \tilde{e}_{q_K, \mathbf{r}^{(K-1)}}^{(K)}. \end{aligned} \quad (36)$$

□

REFERENCES

- [1] Simon S Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. 2017. Hypothesis Transfer Learning via Transformation Functions. In *Advances in Neural Information Processing Systems*. 574–584.
- [2] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-shot imitation learning. In *Advances in neural information processing systems*. 1087–1098.
- [3] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* 50, 2 (2017), 21.
- [4] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. 2017. Label Efficient Learning of Transferable Representations across Domains and Tasks. In *Advances in Neural Information Processing Systems*. 164–176.
- [5] David Minnen, Charles Isbell, Irfan Essa, and Thad Starner. 2007. Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 601–606.
- [6] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.
- [7] Seiya Satoh, Yoshinobu Takahashi, and Hiroshi Yamakawa. 2017. Validation of equivalence structure incremental search. *Frontiers in Robotics and AI* 4, 2 (Dec. 2017). <https://doi.org/10.3389/frobt.2017.00063>
- [8] Seiya Satoh and Hiroshi Yamakawa. 2017. Incremental extraction of high-dimensional equivalence structures. In *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 1518–1524.
- [9] Michael Sipser. 2006. *Introduction to the Theory of Computation*. Vol. 2. Thomson Course Technology Boston.
- [10] Rui Xu and Donald Wunsch. 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks* 16, 3 (2005), 645–678.
- [11] Chin-Chia Michael Yeh, Nickolas Kavantzias, and Eamonn Keogh. 2017. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *2017 IEEE 17th International Conference on Data Mining (ICDM)*.
- [12] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 1317–1322.
- [13] Martijn JA Zeestraten, Ioannis Havoutis, Joao Silvério, Sylvain Calinon, and Darwin G Caldwell. 2017. An approach for imitation learning on Riemannian manifolds. *IEEE Robotics and Automation Letters* 2, 3 (2017), 1240–1247.