Matching Seqlets: An Unsupervised Approach for Locality Preserving Sequence Matching

Jiayan Qiu, Xinchao Wang*, Pascal Fua, *Fellow, IEEE* and Dacheng Tao, *Fellow, IEEE*

Abstract—In this paper, we propose a novel unsupervised approach for sequence matching by explicitly accounting for the locality properties in the sequences. In contrast to conventional approaches that rely on frame-to-frame matching, we conduct matching using *sequencelet* or *seqlet*, a sub-sequence wherein the frames share strong similarities and are thus grouped together. The optimal seqlets and matching between them are learned jointly, without any supervision from users. The learned seqlets preserve the locality information at the scale of interest and resolve the ambiguities during matching, which are omitted by frame-based matching methods. We show that our proposed approach outperforms the state-of-the-art ones on datasets of different domains including human actions, facial expressions, speech, and character strokes.

Index Terms—Sequence Matching, Unsupervised Methods, Temporal Clustering, Joint Optimization

1 Introduction

Sequence matching is of crucial importance to many tasks in computer vision, speech analysis and human computer interaction such as action classification, audio recognition and video retrieval. Given a pair of sequences, the goal of sequence matching is to establish the correspondences between different parts of the two sequences, which provides vital clues for subsequent tasks like sequence classifications.

Despite this straightforward problem definition, sequence matching turns out to be a very challenging problem because of the three main reasons among many. First, for different sequences of the same class, the temporal distributions of the incidents may vary a lot. For example, when speaking the word "eleven", some people spend more time on "e-" and less on "-leven" while others do the opposite; when performing the action "standing up", senior people may act slower than the young. In the case of periodic behaviors, the problem becomes even more demanding due to the repetitive patterns that may occur at different time instants.

Second, consecutive frames in local neighborhoods are often correlated rather than independent of each other. Recall that a frame is merely the consequence of temporal discretization but not a natural entity. Therefore, consecutive frames often share strong similarities and a group of them correspond to one incident in the sequence. If we ignore such affinity and coherence among local frames and only use individual frames for matching, the results can be prone to errors because of the ambiguities between them.

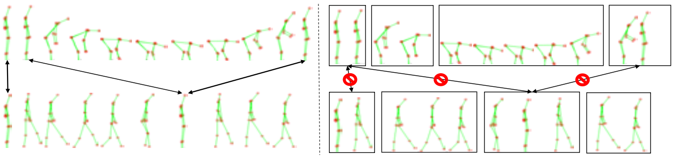

(a) (b)

Fig. 1: Matching two human action sequences using (a) individual frames and (b) seqlets. A seqlet is denoted as a bounding box enclosing skeletons. The upper row depicts a "pick up" action while the lower depicts a "walking". They belong to different classes and therefore fewer matchings, optimally zero, are desirable. Frame-to-frame approach in this case produces three matchings while our seqlet-based one produces none.

Third, micro-level matching labels, such as frame-level ones, are usually too effort-consuming to obtain as it requires significant amount of annotations. Moreover, such matchings are often ambiguous, thus making the human annotations subjective. As a result, sequence classification accuracy is usually adopted as an evaluation measure for sequence matching, despite the two tasks are intrinsically different.

The lack of micro-level labels makes the fully-supervised matching often intractable. Researchers have therefore resorted to unsupervised or weakly-supervised approaches. Existing unsupervised sequence matching approaches, however, focus on alleviating the first problem but barely the second. In other words, they have mechanisms to handle divergent temporal distributions of incidents within the sequences, but do not account for the coherence among frames in the local neighborhoods. For example, Dynamic Time Warping (DTW) [28], the most popular unsupervised sequence matching approach, treats neighboring frames independently, and conducts matching by strictly preserving the temporal orders of the frames, meaning that frames in one sequence can be matched to those in the other only in the same order without any exception. Optimal Transport (OT) [39] and its extension Sinkhorn Distance [8] look at frame-to-frame matching and do not take temporal order into account. The recently proposed Order-preserving

- *J. Qiu and D. Tao are with the UBTECH Sydney Artificial Intelligence Centre and School of Computer Science, Faculty of Engineering and Information Technologies, University of Sydney, NSW, Australia. E-mails: jqiu3225@uni.sydney.edu.au, dacheng.tao@sydney.edu.au.*
- *X. Wang is with the Department of Computer Science, Stevens Institute of Technology, NJ, USA. E-mail: xinchao.wang@stevens.edu. * X. Wang is the corresponding author.*
- *P. Fua is with the School of Computer and Communication Science, EPFL, Lausanne, Switzerland. E-mail: pascal.fua@epfl.ch.*

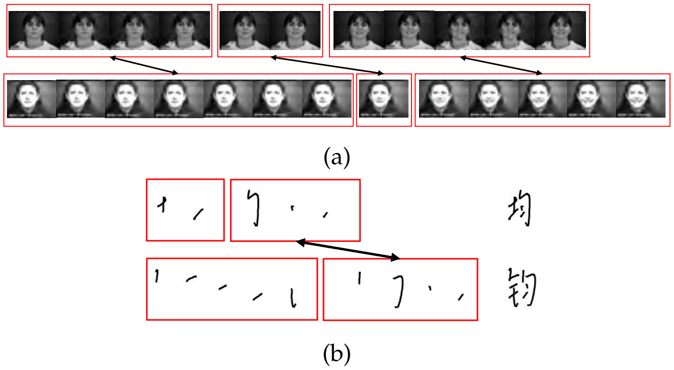
Wasserstein Distance (OPW) [36], combines the flexibility of OT and order-preserving alignment by imposing temporal constrains on OT-based distance, but still neglects the intra-sequence affinities. Existing weakly-supervised approaches like HMM-based ones [33], [34] and LSTM-based ones [9], on the other hand, rely on sequence-level supervisions and also ignore the second problem. Furthermore, unlike unsupervised approaches, such models are often trained explicitly for one application, making them prone to errors when generalized to other application scenarios.

We propose in this paper a fully unsupervised sequence matching approach, without annotations at any level, that explicitly handles all the aforementioned three challenges. The essential idea is to conduct unsupervised sequence matching by clustering neighboring frames into groups of different sizes, which we call *sequencelets'* or *seqlets*, and find the correspondences between seqlets instead of individual frames. Each seqlet is a collection of neighboring frames that share strong affinities and thus assumed to be homogeneous, providing mid-level cues for matching. We thus name our method *Seqlet-Based Matching* (SBM). Our model simultaneously optimizes matching and clustering, which benefit each other, and does not require any supervision from users. The learned seqlets account for the varying temporal spans of events and preserve the local structures for matching at the scale of interest. They also help to resolve the ambiguities especially those erroneous matchings between similar frames from sequences of different classes, yielding results better than the state-of-the-art approaches.

We show in Fig. 1 an example of matching two human action sequences, a "picking up" and a "walking". The two sequences are of different labels and thus fewer matchings, optimally zero, are desirable. Frame-wise matching only accounts for affinities between individual frames across sequences and in this case assigns three matchings, as shown in Fig. 1a, where the matched pairs of skeletons indeed appear very similar. If a wider range of frames are taken into account, as done in our method shown in Fig. 1b, such ambiguities can be removed. Our method simultaneously clusters homogeneous frames to scales of interest and conducts matching on top of clusters, and in the case leads to zero matching between the two sequences. In Fig. 2, we show our matching results on facial expression sequences and online Chinese character stroke sequences. Notably, in the case of character strokes, our approach automatically clusters the strokes of the right parts of the two characters, which are identical, into groups, and match them correctly.

Specifically, our approach starts by computing a number of raw keyframes in each sequence, and then carries out the joint segmentation and matching by taking into account the affinities between seqlet candidates within the same sequence and across the two. We allow for the temporal disorders during matching, as done in OPW, with a cost depending on the number of "crossings". The joint optimization is modeled as a Quadratic Integer Program (QIP) and solved using off-the-shelf solvers. Based on the obtained seqlets, we re-initialize the keyframes and conduct the joint optimization again. The whole process is iterated until convergence.

Our contribution is therefore, to our best knowledge, the first unsupervised model that jointly conducts matching and locality-preserving temporal segmentation of sequences, which is in contrast to prior approaches that ignore such affinities between neighboring frames. Our approach does not require any supervision, and generalizes well to different domains including human actions, facial expressions, speech, and character strokes. We show that our approach yields results superior to those of the state of the art on standard datasets in these domains.



(a)

(b)

Fig. 2: Examples of SBM matching. (a) Matching two *happy* expression sequences. (b) Matching two similar Chinese characters. The left parts of the two characters are different but the right parts are same. SBM automatically groups the strokes of the right part and match them.

2 Related Work

Here we give a brief review of the related work. We start by reviewing the sequence matching methods and then discuss action recognition, one of its direct applications. Note that in this paper we focus on unsupervised sequence matching. Despite supervised methods such as HMM-based ones [33], [34] and LSTM-based ones [9] yield promising results, they still rely on annotations, while ours require no annotations and thus are expected to generalize better to many application domains, as demonstrated later in our experiment section.

2.1 Sequence Matching

DTW is the most commonly used approach for unsupervised sequence matching. Some research work focus on reducing the high computation cost of DTW. The Fast-DTW [29] and SparseDTW [1] are two modified versions of DTW that run faster under certain circumstances. Silva et al. [31] reduce the computation cost of DTW by speeding up the calculation of its all-pairwise DTW matrix, and Ratanamahatana et al. [25] improve the efficiency of DTW by constraining the amount of warping. Canonical Time Warping [48] and generalized time warping [47] generalize DTW to handle sequences with changing dimension frames. Besides, the Longest Common Subsequences distance [5] and the edit distance [22], which are designed for string comparisons, are extended to deal with the multi-dimensional sequences matching in [21] and [40].

Some recent approach focus on learning-based matching but still rely on DTW for computing distances. For example, the approach of [26] generates a new sequence first by mapping the original sequence to the learned semi-continuous HMMs, and then extracts the mixture weights of states. This is achieved under the supervision of sequence-level labels. The DTW distance between the weight sequences is then treated as the distance between sequences. Su et al. [33], [34] extract the HMM-based statistic information from each set of sequences. Then the DTW distance between sequence statistic information is used as distance between sequences. A rate-invariant distance based on the transported square-root vector field representation is proposed in [38] and applied on action recognition tasks [2]. Since these methods all rely on DTW, they suffer the same problem as DTW.

Apart from DTW, some other distances have also been adopted for sequence matching. Garreau et al. [11] propose to learn Mahalanobis distance for temporal sequence alignment. However, the true alignment of the learning examples must be known a priori. Su et al. [36], on the other hand, propose an OT-based method, where two temporal regularization terms are added to preserve the temporal information of sequences in the matching process.

Unlike existing sequence matching methods that focus on frame-to-frame matching, our method conducts seqlet-to-seqlet matching, where locality is preserved and seqlets are automatically learned without any human supervision.

Note that the sequence matching task is very different from the sequence classification one, whose goal is to classify a sequence into one of the existing types. Nevertheless, the latter task is often used as an evaluation measure for the former as done in our experiments, due to the lack of frame-to-frame matching annotations.

2.2 Action Recognition

Action recognition is one of many direct applications of sequence matching. For this task, various graph-based models, such as the spatio-temporal graph [6], the temporal AND-OR graph [23], [46] and the actom sequence models [10] have been used. Sadanand et al. [27] embed the temporal information in the activity representation, while Anirudh et al. [3] propose a geometry and data adaptive symbolic framework to improve the efficiency of action sequence recognition. The approach of [17], performs isolated recognition based on per-frame representation of sequence, and on aligning test sequence with its model sequence. The approaches of [41], [42], [45], on the other hand, use the trajectory information on action sequence recognition. The ones of [18], [19], [30] map action sequence on an HMM model to carry out sequence prediction. Jiang et al. [14], [15] convexity the action matching problems into a linear programming task. Recently, Su et al. [37] parse action sequences hierarchically into segments using the temporal information, and have achieved very promising performances. Also, deep learning models have been implemented for action recognition [7], [13], [32] and have achieved promising performance.

Our proposed seqlet matching approach can be directly applied to not only action recognition, but also other domains like facial expressions, speech and character strokes.

3 Model

Our method conducts sequence matching using seqlets, which explicitly takes into account the local affinities between consecutive frames within each sequence. Specifically, our method jointly optimizes the intra-sequence seqlet selection and inter-sequence seqlet matching. This is achieved by our formulation of the problem as a Quadratic Integer Program (QIP). In what follows, we first go through the workflow of our method, with the visual illustration shown in Fig. 3, then give the definitions of individual components, and finally show the complete QIP formulation with constraints.

Our method works in an iterative manner by repeating the keyframe extraction and seqlet matching until convergence. We start by extracting initial keyframes in both sequences, for which the details will be provided in Sec. 4.1. Based on the obtained keyframes, we construct seqlets candidates, shown as the black bounding boxes in Fig. 3. Initially, each seqlet candidate is formed by including only one keyframe and the possible non-keyframes. We then allow at most three consecutive seqlet candidates to join together into a longer one, which we name as the *merging* process. For a merged seqlet, we re-compute a new keyframe f_{i^*}, taken to be the one whose sum of distances to all the frames is minimized:

$$i^* = \arg\min_{i \in I} \sum_{t \in I} \|f_t - f_i\|_2 \,, \tag{1}$$

where I is the set of indices of all the frames contained in the seqlet, and $\|\cdot\|_2$ denotes the L2 norm. In this way, each seqlet, merged or not, comprises only one keyframe.

We model the joint seqlet selection and matching using a graph shown in Fig. 3, where we treat a seqlet as a node, and a possible link between a pair of seqlets as an edge. On top of each edge, we define a binary variable, indicating whether the edge is selected or not in the final solution. We categorize the variables into two types as follow:

- $h_{i,j} \in \{0, 1\}$, for seqlet selection, defined on edges between Seqlets i and j within a sequence. $h_{i,j} = 1$ indicates the two seqlets are chosen and linked.
- $s_{i,j} \in \{0, 1\}$, for seqlet matching, defined on edges between Seqlet i of the first sequence and Seqlet j of the second. $s_{i,j} = 1$ indicates the two seqlets are matched.

In Fig. 3, we denote $h_{i,j}$ in blue and $s_{i,j}$ in yellow. Note that, we define an edge between two seqlets in the same sequence only if they are neighbors, meaning that they are temporally consecutive without any overlap or gap.

Given the keyframes and thus the constructed graph, we model our joint optimization problem as a QIP of the selection variables. We write our objective function as

$$\min_{\mathbf{H},\mathbf{S}} E(\mathbf{H}, \mathbf{S}) = \min_{\mathbf{H},\mathbf{S}} E_m(\mathbf{S}) + E_c(\mathbf{H}) + E_d(\mathbf{S}) + E_l(\mathbf{S}), \tag{2}$$

where $\mathbf{H}$ denotes the set of all $h_{i,j}$ and $\mathbf{S}$ denotes the one of all $s_{i,j}$. E_m, E_c, E_d, and E_l denote the energy terms corresponding to the seqlet matching, clustering, crossing penalty and length penalty, respectively.

We solve the optimization of Eq. (2) and denote the optimal solutions as $\mathbf{H}^*$ and $\mathbf{S}^*$, which correspond to the optimal seqlet selection and matching under the current keyframe setting. In Fig. 3, we use the red edges to denote the selected $h_{i,j}$ and black edges to denote the selected $s_{i,j}$. We then use the obtained seqlets to compute the new

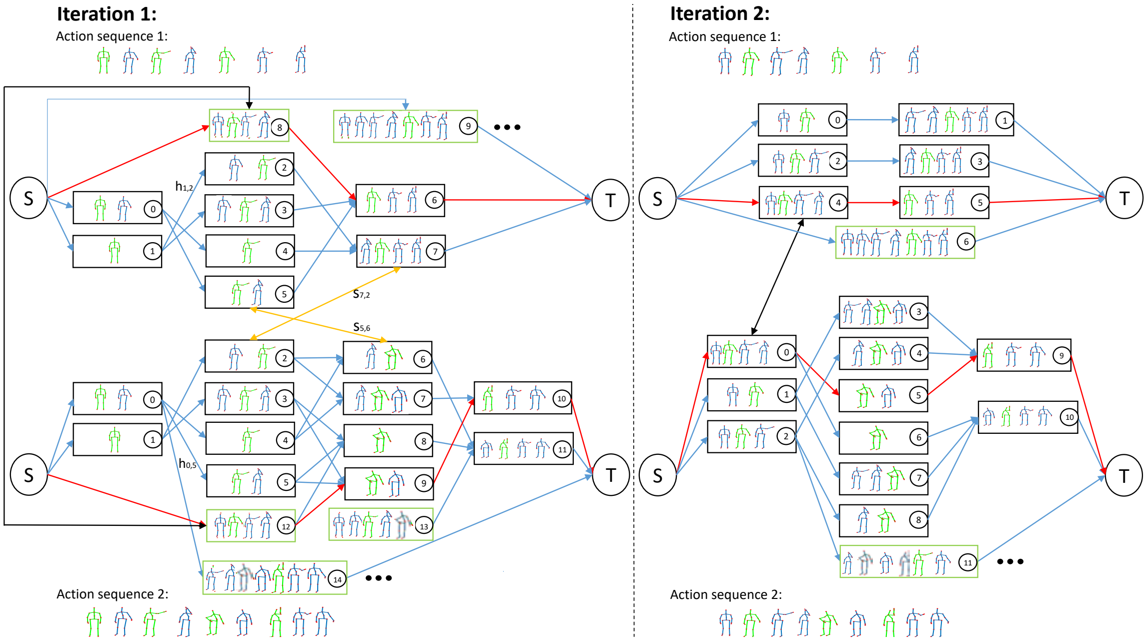


Fig. 3: Illustration of our matching process. We show two action sequences that comprise seven and nine frames respectively. We use a black bounding box to denote a seqlet, and a green box to denote a merged one. Each selection edge h_{ij} is colored blue, and each matching edge s_{ij} is colored yellow. Note that, for a more concise presentation, we show only a pair of matching edges. A green skeleton denotes a keyframe. Also, we highlight a selected h_{ij} using red and a selected s_{ij} using black. By the end of the first iteration, Seqlets 8 and 6 are selected for sequence 1, and Seqlets 12, 9, and 10 are selected for sequence 2. Meanwhile, Seqlet 8 of sequence 1 is matched with Seqlet 12 of sequence 2. In Iteration 2, we update the keyframes and conduct the joint clustering-matching again. This process is repeated until convergence.

keyframes using Eq. (1), and iterate the above process. We stop the iterations when the objective of Eq. (2) stops decreasing. In what follows, we give details of the four energy terms in Eq. (2).

3.1 Matching Term

The matching term E_m measures the similarity between a pair of seqlets across two sequences. We take it to be

$$E_m(\mathbf{S}) = \sum_{i,j:j\in\mathcal{M}(i)} c_{i,j}s_{i,j}, \tag{3}$$

where $c_{i,j}$ encodes the distance-based cost between Seqlet i in the first sequence and Seqlet j in the second, $\mathcal{M}(i)$ denotes the set of all seqlets in the other sequence that can be potentially matched to Seqlet i, and $s_{i,j}$, as discussed, is a binary variable denoting the matching between Seqlets i and j. To compute $c_{i,j}$ between two seqlets, we need to account for their different lengths. In our implementation, we up-sample the short seqlet by interpolation so that it has the same number of frames as the long one, and then compute the distance-based cost between the two equal-length sequences. Note that the $c_{i,j}$, whose details will be provided in Sec. 4.2, takes a negative value when seqlets are similar and a positive one when their distance is large.

3.2 Clustering Term

The clustering term E_c accounts for the grouping of consecutive frames within the same sequence. Intuitively, frames with similar features are more likely to correspond to the same incident and thus should be clustered into a seqlet. We model each possible clustering or seqlet candidate as a node in our graph, and then use the binary variable h_{ij} to denote the linking of Seqlets i and j, where $h_{ij} = 1$ indicates that both seqlets are selected. We define E_c to be

$$E_c(\mathbf{H}) = \alpha_1 \cdot \frac{1}{K} \sum_{i,m:m\ \in\mathcal{N}(i)} e_{i,m}h_{i,m}, \tag{4}$$

where $e_{i,m}$ encodes the affinities among frames within Seqlets i and m, K is the number of keyframes in the sequence. $\mathcal{N}(i)$ is the set of all neighbors of Seqlet i, and α_1 is a weighting factor to balance the energy terms. Note that, we define $m\ \in \mathcal{N}(i)$ only if the first frame of Seqlet m is right next to the last frame of Seqlet i. We take the cost $e_{i,m}$ to be

$$e_{i,m} = \frac{1}{2}\left(\frac{1}{F_i}\sum_{t\in I_i}\|f_t - f_i^*\|_2 + \frac{1}{F_m}\sum_{t\in I_m}\|f_t - f_m^*\|_2\right), \tag{5}$$

where F_i and F_m denote respectively the total number of frames in Seqlets i and m, f_i^* and f_m^* denote their corresponding keyframes, while I_i and I_m denote respectively the sets of all frame indices in Seqlets i and m.

3.3 Crossing Penalty Term

We allow for non-chronological matching of seqlets across two sequences, as incidents may not follow the exact same order even for sequences of the same type. We impose a cost for such matching that depends on the number of non-chronological matchings. Specifically, if two seqlets in the first sequence match the other two in the second sequence in a reverse-time order, the two matching edges, each of which links two seqlets in the two sequences, lead to a "crossing", which is penalized in our objective. This means the more distant apart two seqlets are, the more crossing it costs for their matching. We write the crossing penalty term E_d as

$$E_d(\mathbf{S}) = \alpha_2 \cdot \sum_{\substack{i,j,m,n:\\ j\in\mathcal{M}(i)\\ m\in\mathcal{M}(n)}} d^{i,m}_{j,n} s_{i,j} s_{m,n}, \tag{6}$$

where $d^{i,m}_{j,n}$ is the cost of "crossing" between $s_{i,j}$ and $s_{m,n}$, and α_2 is the weight. We take $d^{i,m}_{j,n} = 1$ for crossed edges, i.e., $(t\,(i) - t\,(m)) \cdot (t\,(j) - t\,(n)) < 0$ where $t\,(i)$ denotes the starting time of Seqlet i, and otherwise we take $d^{i,m}_{j,n} = 0$.

3.4 Inverse Length Penalty Term

We also introduce a term, E_l, to encourage longer seqlets to be selected. This is because the selections of longer seqlets tend to produce fewer number of matched edges, which potentially lead to larger objective values and are thus impeded by the optimization. We therefore penalize the inverse length of seqlets by defining

$$E_l(\mathbf{S}) = \alpha_3 \cdot \sum_{i,j:j\in\mathcal{N}(i)} l_{i,j} s_{i,j}, \tag{7}$$

where $l_{i,j} = \frac{1}{2}(\frac{1}{l(i)} + \frac{1}{l(j)})$, $l(i)$ denotes the temporal length of Seqlet i, and α_3 is the weight.

3.5 Constrained QIP

We have now described each term in our objective function $E(\mathbf{H}, \mathbf{S})$ shown in E.q. (2), where all the variables to be optimized, $\mathbf{H}$ and $\mathbf{S}$, are binary. However, not all assignments of $\mathbf{H}, \mathbf{S}$ are physically plausible: they should obey some hard constraints so that the intra-sequence clustering and inter-sequence matching can influence and interact with one another in a positive way. In our complete QIP model, we include the following constraints.

The first set of constraints enforce that within a sequence, each seqlet can be selected at most once:

$$\sum_{m\in\mathcal{N}(i)} h_{i,m} \leq 1, \forall i. \tag{8}$$

To handle the seqlets in the beginning and at the end of a sequence, we introduce a virtual source node S and a sink node T. For each source S, we set $\sum_{m\in\mathcal{N}(S)} h_{S,m} = 1$, which ensures that one and only one path of nodes can be selected within each of the two sequences.

The second set of constraints implement the "flow conservation", meaning that each seqlet, if selected, must be linked to two other neighboring seqlets, a left neighbor and a right one. We link S to all the seqlets starting with the first frame, and link all the seqlets ending with the last frame to T. We write

$$\sum_{i:m\in\mathcal{N}(i)} h_{i,m} = \sum_{k\in\mathcal{N}(m)} h_{m,k}, \forall m. \tag{9}$$

The third set of constraints enforce that, each seqlet can be matched only if it is selected. In other words, for each seqlet, the sum of all the matching variables should be less or equal to the sum of all selection variables. We write

$$\sum_{j\in\mathcal{M}(i)} s_{i,j} \leq \sum_{m\in\mathcal{N}(i)} h_{i,m}, \forall i, \tag{10}$$

$$\sum_{j:i\in\mathcal{M}(j)} s_{i,j} \leq \sum_{n\in\mathcal{N}(j)} h_{j,n}, \forall i, \tag{11}$$

where Constraint (10) accounts for Seqlet i in the first sequence and Constraint (11) accounts for Seqlet j in the second.

Our QIP is therefore a program with an objective function of (2) and with constraints of (8), (9), (10), and (11). We solve this QIP using Gurobi, a state-of-the-art commercial solver. The obtained assignments of $\mathbf{H}$ and $\mathbf{S}$ variables indicate the optimal frame clustering for matching, and the resulting objective indicates the minimum distance between the two sequences. We keep record of the obtained objective value, re-initialize the keyframes, re-run the QIP optimization, and iterate this process until the objective value stops decreasing.

4 Implementation Details

We provide here the implementation details of our keyframe extraction and distance-based cost between seqlets.

4.1 Keyframe Extraction

Our initial keyframe or the keyframe in the first iteration is obtained as follows. We start by setting the first frame in a sequence to be a keyframe, and then loop each frame and check its distance to the previous keyframe using two criteria, which we give details below. If this distance is larger than a threshold, we set this frame to be a new keyframe. We go through each frame in this way until we reach the end of the sequence.

The first criterion concerns the normalized difference between the current frame and the last keyframe. We define $\lambda_1 = \sum_{m=1}^{N} \left| \frac{f^c_m - f^k_m}{f^k_m} \right|$ and $\lambda_2 = \sum_{m=1}^{N} \left| \frac{f^c_m - f^k_m}{f^c_m} \right|$, where N is the dimensionality of features, f^c_m is the m-th dimension feature of the current frame and f^k_m is the m-th dimension feature of the last keyframe. Therefore, λ_1 and λ_2 measure the relative changes in features with respect to the last keyframe and the current frame, respectively.

The second criterion regards the feature with the largest relative change. In human action matching, for example, $wave\ hand$ leads to only local changes in the features corresponding to the locations of hands and arms. We write the change over a single feature dimension as $\lambda_3 = \max_{m=1,\ldots,N} \left| \frac{f^c_m - f^k_m}{f^k_m} \right|$ and $\lambda_4 = \max_{m=1,\ldots,N} \left| \frac{f^c_m - f^k_m}{f^c_m} \right|$, where again f^c_m and f^k_m are the m-th dimension feature of the current frame and the last keyframe respectively. Finally, we set a frame to be a keyframe if $(\lambda_1 > \beta_1) \vee (\lambda_2 > \beta_1) \vee (\lambda_3 > \beta_2) \vee (\lambda_4 > \beta_2)$, where $\vee$ is the **or** logical operation, and β_1 and β_2 are hand-set thresholds. In practice, β_1 and β_2 are set lower so that in the first iteration we produce an over-complete set of keyframes and thus a dense initial temporal segmentation.

In the following iterations, we extract one keyframe from each obtained seqlet using Eq. (1). Intuitively, this means that we select the most "representative" frame in the least square sense within each seqlet, and use it for the next iteration of matching.

4.2 Distance-Based Cost Between Seqlets

As we test our method on different domains, we have to re-calibrate the distances so as to obtain plausible results. For matching problems that require pre-alignment, such as the human actions and facial expressions to be introduced in Sec. 5, we take the distance between seqlets as $d = \min_{T,R,s} \frac{1}{N} \|sR(M_1 + T) - M_2\|_F$, where M_1 and M_2 are the two seqlets after upsampling, N is the number of elements of M_1, T is the translation matrix, R is the rotation matrix, s is the scale parameter, and $\|\cdot\|_F$ denotes the Frobenius norm. For aligned data, like in the Spoken Arabic Digits (SAD) and Similar Online Chinese Character datasets to be described in Sec. 5, we take the distance to be $d = \frac{1}{N} \|M_1 - M_2\|_F$. Based on the obtained distance d, we then compute our cost c_{ij} as $c_{ij} = \frac{e^d-1}{e^d+1} - \frac{1}{2}$, so that c_{ij} can be both negative and positive.

5 Experiments

We test our proposed SBM on five benchmarks of different domains: human action datasets including MSR-Action3D [19] and MSRDailyActivity3D [43], speech dataset Spoken Arabic Digits (SAD) [4], facial expression dataset Cohn-Kanade [20], and character stroke dataset Similar Online Chinese Character [34], [35]. In what follows, we show our comparative results on the five datasets and provide experimental analysis.

5.1 Evaluation Measures

We evaluate the matching performance in terms of sequence classification accuracies, as done in [16], [24], [36], since there are currently no large-scale publicly-available datasets with frame- or seqlet- level matching ground truths. We apply two classifiers, the *k* nearest neighbor (k-NN) classifier and the nearest mean (NM) classifier. The k-NN classifier classifies a test sample based on majority voting among the k nearest neighbors, and the NM one assigns a sample based on distance between the test sample and the centroid of each class. We use the mean average precision (MAP) and the classification accuracy as our evaluation measures, both of which are computed based on the ranking of the distance between the test sample and the training ones.

5.2 Baselines Methods

We compare our proposed method, SBM, with a number of unsupervised sequence matching methods including the state-of-the-art ones listed as follows.

- **DTW** [28]: It is most popular sequence matching method, which strictly preserves the temporal order of the frames during matching.
- **nDTW** [12]: It is a normalized version of DTW, which accounts for the matching steps.
- **Sinkhorn Distance** [8]: It is a smoothed and computationally efficient version of Optimal Transport, providing a canonical method to lift the geometry between instances so as to compute the distance between sequences.
- **OPW** [36]: It is the current state-of-the-art method in sequence matching. It is a variation of Optimal Transport and computes the Wasserstein distance between two sequences.

Note that, the results of SBM is obtained using the best parameters obtained using 5-fold validation.

5.3 MSR-Action3D Dataset

The MSR-Action3D (MSR-3D) Dataset [19] comprises 23,797 frames from 567 action sequences. It features 20 sports actions, each of which is performed by 10 subjects two to three times. We split the dataset into the training and testing sets according to the subjects, as done in [36], [43], [44], where the action sequences performed by five of the ten subjects are used for training and those by the other five are used for testing. We use the raw skeleton joint positions as features for SBM. We set α_1, α_2 and α_3 to be 0.01, 0.01 and 0.05 respectively, and set β_1 and β_2 to be 1.5 and 0.20.

We show the results in Fig. 4a and 4f. As can be seen, our proposed methods SBM beats all other methods using all the metrics. SBM outperforms the state-of-the-art method, OPW, by 0.98% on MAP and 2.61% on 1-NN.

5.4 MSRDailyActivity3D Dataset

The MSRDailyActivity3D (MSRD-3D) dataset [43] consists of 16 activity types, where each type is performed by 10 subjects twice using two different poses: *standing* and *sitting on the sofa*. We split the dataset by setting half of the 10 subjects to be the training set and the other half for testing, as done in [36], [43], [44]. We again use the joint positions as features for SBM. In this experiment, all the parameters, i.e., α_1, α_2, α_3 and β_1 and β_2, are set to be the same values as in the MSR-Action3D Dataset.

The results are shown in Fig. 4b and 4g. Our proposed SBM achieves the best results in five out of six evaluation measures. The MAPs of the NM classifier for all the compared methods are low. This is because the representation sequence of NM is always far away from part of the test sequences in the same class, since the same activity is performed with both *standing* and *sitting on the sofa*. For 1-NN and 3-NN, SBM outperforms OPW distance by 2.50% and 5.01%, despite that the difference between SBM and OPW is 0.65% for 5-NN. For the NM and MAP of NM, SBM beats the best method by 1.25% and 0.93%. The results show that the overall performance of SBM outperforms other sequence matching methods.

5.5 Spoken Arabic Digits Dataset

The SAD dataset includes 10 spoken Arabic digits from 88 subjects, where each subject speaks a digit for 10 times. Among the 88 subjects, half of them are female and half are male. In the experiment, 660 samples of each digits and thus in total 6,600 audio sequences are used for training, and the other 2,200 sequences are used for testing, as done in [36]. We use the mel-frequency cepstrum coefficients (MFCCs) provided by the author of [4] as the feature. In this experiment, the parameters α_1, α_2 and α_3 are set to be the same as in MSR-Action3D and MSRDailyActivity3D, but β_1 and β_2 are set to be 10 and 5 due to the feature scales.

It can be seen from Fig. 4c and 4h that SBM outperforms all the compared methods, although the margin is not significant. This is because the SAD dataset consists of only audio sequences of digits 0 to 9, where each digit contains limited syllables, making the recognition task simple.

5.6 Cohn-Kanade Dataset

The Cohn-Kanade (CK) dataset comprises 7 different emotional expressions from 123 subjects. Although it contains in total 593 sequences, only 327 emotional expression sequences from 118 subjects are labeled and can be used for our evaluation. In our experiment, we use 166 sequences from 59 subjects for training and the rest 161 sequences from the other 59 subjects for testing. The AAM landmarks are used as features. We set α_1, α_2 and α_3 to be the same values as in MSR-Action3D, and set β_1 and β_2 to be 1.00 and 0.01.

We show in Fig. 4d and 4i the comparative results on the CK dataset, where the proposed SBM outperforms all the compared ones including the state-of-art method OPW [36] on all evaluation metrics.

5.7 Similar Online Chinese Character Dataset

The Similar Online Chinese Character (SOCC) dataset includes 7 similar Chinese characters and each have 107 different samples. In this experiment, we use 54 out of 107 samples as training set and the other samples as testing set. Different with [34], [35] that using a 10-dimensional feature for each stroke, we directly use the points recorded with each character as the input data in our experiment. For this dataset, we set α_1, α_2 and α_3 to be 0.0005, 0.01 and 0.5 respectively. Moreover, different with the initial keyframe extraction manner of previous experiments, we extract the initial keyframes by using the absolutely difference between data, and we set β_1 and β_2 to be 5 and 3.

As can be seen from Fig. 4e and 4j, SBM again outperforms all the compared algorithms on all evaluation metrics. Although the 3-NN accuracy of SBM only obtains an improvement of 0.11% compared to nDTW, the overall performance of SBM is much better.

5.8 Sensitivity of Parameters

In Tab. 1, we show the influence of parameters, α_1, α_2, α_3 and the (β_1,β_2) pair, on the matching results. We record the results by varying one parameter while freezing the others.

Tab. 1a shows that when α_1 is large, the matching accuracy decreases as α_1 increases. This is because when the cluster term is heavily weighted, the seqlets tend to include as few frames as possible. Also, when α_2 is small, the accuracy increases as α_2 increases; when α_2 is large, the accuracy decreases. This is because when the weight of crossing penalty term is small, temporal order information tends to be ignored. When the crossing penalty is large, however, SBM becomes a DTW-like matching, leading to the problem of sparse matching. When α_3 is small, the accuracy increases with the increasing of α_3, in which case the inverse length penalty term guides the optimization to find more suitable seqlets. However, when α_3 is too large, the optimization will result in long seqlets, making SBM neglect locality temporal information.

We show in Tab. 1b the influence of the pair (β_1, β_2), which are thresholds to extract the initial keyframes. Smaller values of (β_1, β_2) lead to more initial keyframes and thus more iterations to converge, but more chances to capture the optimal seqlets and thus the better performance.

α_1	**0.005**	**0.010**	**0.020**	**0.030**	**0.040**	**0.050**
Accuracy	86.20	86.86	86.86	86.86	86.53	86.53
α_2	**0.005**	**0.010**	**0.020**	**0.030**	**0.040**	**0.050**
Accuracy	83.16	86.86	86.20	85.81	85.12	83.50
α_3	**0.02**	**0.05**	**0.100**	**0.200**	**0.300**	**0.500**
Accuracy	85.52	86.86	85.86	85.12	84.42	80.47

(a)

(β_1,β_2)	**(0.0, 0.00)**	**(0.5,0.10)**	**(1.0,0.15)**	**(1.5,0.20)**	**(2.0,0.30)**
Accuracy	86.86	86.86	86.86	86.86	80.47

(b)

TABLE 1: The influence of α_1, α_2, α_3 and (β_1,β_2) on the MSR-Action3D dataset.

5.9 Influences of SBM Terms

Here we test the influences of the terms in SBM, by turning one or more terms off and then comparing their corresponding performances. The experiments are conducted on the MSR-3D dataset. Our complete model adopts the parameter setting in Sec. 5.3. We first show the performance with one term turned off, then with two terms, and finally with all the three terms turned off. It can be seen that the performance drops when removing more terms, indicating that each term does play a role in positively influencing SBM, and all terms together yield the promising performance of SBM.

Parameters			**Accuracy**
α_1	α_2	α_3	
0	0.01	0.05	80.80
0.01	0	0.05	79.12
0.01	0.01	0	75.08
0.01	0	0	72.05
0	0.01	0	74.41
0	0	0.05	73.74
0	0	0	72.05
Complete Model			86.86

TABLE 2: The influences of the terms in SBM.

6 Conclusion

In this paper, we propose a novel unsupervised sequence matching approach, named Seqlet-Based Matching (SBM). In contrast to conventional frame-to-frame matching methods, ours computes the correspondences between groups of homogeneous frames, which we name as seqlets. Our method looks at a longer and dynamic temporal range of frames for matching and thus helps to remove the ambiguities of frame-based associations. The optimal sets of seqlets and matching are learned jointly, without any supervision from human users. We compare SBM with state-of-the-art sequence matching approaches in different domains including human actions, facial expressions, speech, and character strokes, and show that SBM yields superior results.

Acknowledgment

This work is in part supported by the Australian Research Council Project FL-170100117, and the Stevens Institute of Technology Startup Funding.

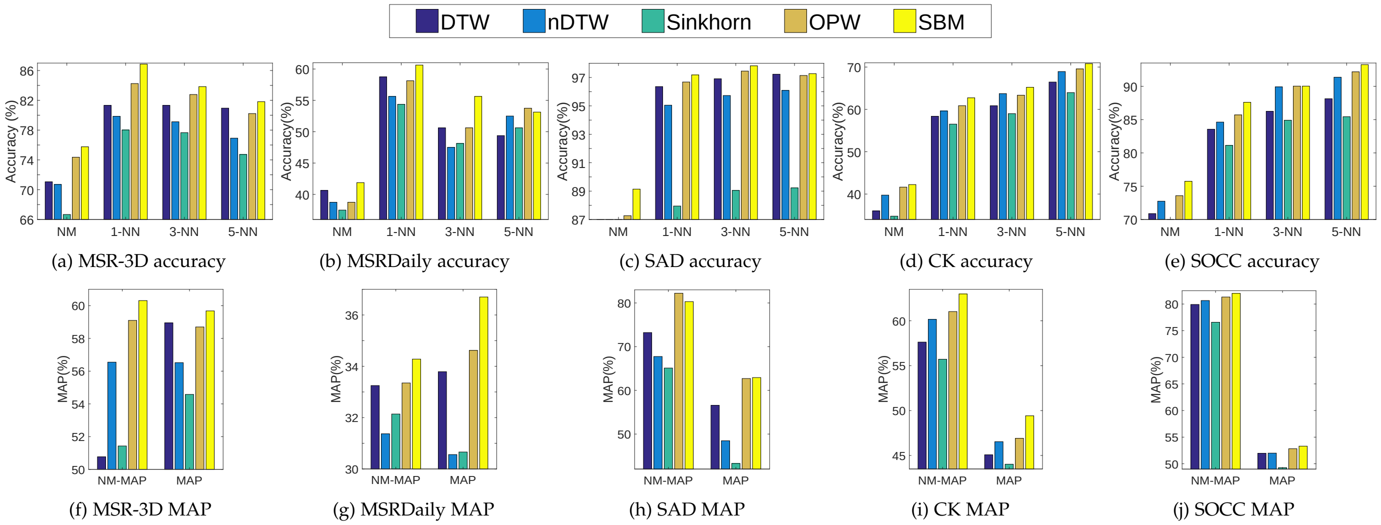
References

[1] G. Al-Naymat, S. Chawla, and J. Taheri. Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101*, 2009.

[2] B. B. Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *TPAMI*, 38(1):1–13, 2016.

[3] R. Anirudh and P. Turaga. Geometry-based symbolic approximation for fast sequence matching on manifolds. *IJCV*, 116(2):161–173, 2016.

[4] K. Bache and M. Lichman. Uci machine learning repository. university of california, irvine, school of information and computer sciences, 2013. *URL: http://archive. ics. uci. edu/ml*, 2013.



(a) MSR-3D accuracy (b) MSRDaily accuracy (c) SAD accuracy (d) CK accuracy (e) SOCC accuracy

(f) MSR-3D MAP (g) MSRDaily MAP (h) SAD MAP (i) CK MAP (j) SOCC MAP

Fig. 4: Comparative results on the MSR-Action3D, MSRDailyActivity3D, SAD, Cohn-Kanade and Similar Online Chinese Character dataset. NM, 1-NN, 3-NN, and 5-NN denote the results evaluated using classification accuracy, while NM_MAP and MAP denote the NM and 1-NN results using MAP, respectively.

[5] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval*, pages 39–48, 2000.

[6] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

[7] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, 2015.

[8] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.

[9] P. Dogan, B. Li, L. Sigal, and M. Gross. A neural multi-sequence alignment technique (neumatch). In *CVPR*, 2018.

[10] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011.

[11] D. Garreau, R. Lajugie, S. Arlot, and F. Bach. Metric learning for temporal sequence alignment. In *NIPS*, 2014.

[12] S. Gudmundsson, T. P. Runarsson, and S. Sigurdsson. Support vector machines and dynamic time warping for time series. In *IJCNN*, 2008.

[13] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013.

[14] H. Jiang, M. S. Drew, and Z.-N. Li. Successive convex matching for action detection. In *CVPR*, 2006.

[15] H. Jiang, M. S. Drew, and Z.-N. Li. Matching by linear programming and successive convexification. *TPAMI*, 29(6):959–975, 2007.

[16] H.-J. Jung and K.-S. Hong. Enhanced sequence matching for action recognition from 3d skeletal data. In *ACCV*, 2014.

[17] K. Kulkarni, G. Evangelidis, J. Cech, and R. Horaud. Continuous action recognition based on sequence alignment. *IJCV*, 112(1):90–114, 2015.

[18] K. Li, J. Hu, and Y. Fu. Modeling complex temporal composition of actionlets for activity prediction. *ECCV*, 2012.

[19] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPR workshop*, 2010.

[20] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPR workshop*, 2010.

[21] P.-F. Marteau. Time warp edit distance with stiffness adjustment for time series matching. *TPAMI*, 31(2):306–318, 2009.

[22] G. Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, pages 31–88, 2001.

[23] M. Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. In *ICCV*, 2011.

[24] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *ICDM*, 2014.

[25] C. A. Ratanamahatana and E. Keogh. Making time-series classification more accurate using learned constraints. In *ICDM*, 2004.

[26] J. A. Rodríguez-Serrano and F. Perronnin. A model-based sequence similarity with application to handwritten word spotting. *TPAMI*, 34(11):2108–2120, 2012.

[27] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[28] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[29] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[30] Z. Si, M. Pei, B. Yao, and S.-C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *ICCV*, 2011.

[31] D. F. Silva and G. E. Batista. Speeding up all-pairwise dynamic time warping matrix calculation. In *ICDM*, 2016.

[32] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*. 2014.

[33] B. Su and X. Ding. Linear sequence discriminant analysis: a model-based dimensionality reduction method for vector sequences. In *ICCV*, 2013.

[34] B. Su, X. Ding, C. Liu, H. Wang, and Y. Wu. Discriminative transformation for multi-dimensional temporal sequences. *TIP*, 26(7):3579–3593, 2017.

[35] B. Su, X. Ding, H. Wang, and Y. Wu. Discriminative dimensionality reduction for multi-dimensional sequences. *TPAMI*, 40(1):77–91, 2018.

[36] B. Su and G. Hua. Order-preserving wasserstein distance for sequence matching. In *CVPR*, 2017.

[37] B. Su, J. Zhou, X. Ding, and Y. Wu. Unsupervised hierarchical dynamic parsing and encoding for action recognition. *TIP*, 26(12):5784–5799, 2017.

[38] J. Su, S. Kurtek, E. Klassen, A. Srivastava, et al. Statistical analysis of trajectories on riemannian manifolds: bird migration, hurricane tracking and video surveillance. *The Annals of Applied Statistics*, pages 530–552, 2014.

[39] C. Villani. *Optimal transport: old and new*, volume 338. 2008.

[40] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multidimensional time-series. *The VLDB Journal*, pages 1–20, 2006.

[41] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.

[42] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[43] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[44] J. Wang and Y. Wu. Learning maximum margin temporal warping for action recognition. In *ICCV*, 2013.

[45] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[46] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010.

[47] F. Zhou and F. De la Torre. Generalized time warping for multi-modal alignment of human motion. In *CVPR*, 2012.

[48] F. Zhou and F. Torre. Canonical time warping for alignment of human behavior. In *NIPS*, 2009.