Pattern Recognition 79 (2018) 1–11

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A graph-based approach for detecting common actions in motion capture data and videos

Costas Panagiotakis[a,c,*], Konstantinos Papoutsakis[b,c], Antonis Argyros[b,c]

[a] Department of Business Administration, Technological Educational Institute of Crete, Crete, Agios Nikolaos 72100 Greece
[b] Computer Science Department, University of Crete, Crete, Heraklion 70013 Greece
[c] Computational Vision and Robotics Laboratory, Institute of Computer Science, FORTH, Crete, Heraklion 70013 Greece

ARTICLE INFO

Article history:
Received 4 August 2017
Revised 25 January 2018
Accepted 2 February 2018

Keywords:
Common action detection
Video co-segmentation
Temporal action co-segmentation
Dynamic Time Warping

ABSTRACT

We present a novel solution to the problem of detecting common actions in time series of motion capture data and videos. Given two action sequences, our method discovers all pairs of common subsequences, i.e. subsequences that represent the same or similar action. This is achieved in a completely unsupervised manner, i.e., without any prior knowledge of the type of actions, their number and their duration. These common subsequences (commonalities) may be located anywhere in the original sequences, may differ in duration and may be performed under different conditions e.g., by a different actor. The proposed method performs a very efficient graph-based search on the matrix of pairwise distances of frames of the two sequences. This search is supported by an objective function that captures the trade off between the similarity of the common subsequences and their lengths. The proposed method has been evaluated quantitatively on challenging datasets and in comparison to state of the art approaches. The obtained results demonstrate that the proposed method outperforms the state of the art methods both in the quality of the obtained solutions and in computational performance.

1. Introduction

The unsupervised discovery of similar temporal patterns (e.g. similar actions) in time series is considered as an important problem in pattern recognition and computer vision. In this research, we are interested in the detection of common subsequences (commonalities) in two sequences of motion capture data or videos of continuous actions, without any prior knowledge on the type of actions, their number, or their duration. The problem was introduced by Chu et al. [1] as Temporal Commonalities Discovery (TCD), applied to pairs of image sequences containing facial expressions or motion capture data. It has also been tackled in [2] as co-action discovery in multiple image sequences, in [3] as video co-segmentation for action extraction and more recently in [4] as temporal action co-segmentation in pairs of videos. A similar problem appearing in the knowledge discovery and data mining community considers the discovery of multiple common patterns within the same signal [5,6], time series [7,8], or string [9]. In this setting, the discovered commonalities are called motifs [7,8]. This work is also motivated by the task of unsupervised discovery of common human actions in this type of input [10]. Other relevant problems include image co-segmentation [11], image/video co-localization [12] and video co-summarization [13].

The discovery of commonalities in time series is a challenging problem with applications in several domains, including but not limited to data mining and content retrieval, audio and natural language processing, image/video analysis, bio-informatics, economics, physics and more. Both the supervised and the unsupervised versions of the problem are of great importance and interest [14]. For example, the detection of the longest common subsequence has been successfully used for dynamic hand gesture classification [15]. The problem of periodicity detection [16,17] can also be seen as an instance of the problem of finding commonalities/common subsequences between two different videos. In addition, the detected commonalities between video pairs can be used in video retrieval [18,19] which is the task of finding the most similar video based on a query video. Action co-segmentation can also be used to detect recurring combinations of actions without knowledge of what the common events are, how many there are, or when they begin and end. When the action labels of one of the two sequences are given, the detection of commonalities can be used for human action recognition [20–27], which constitutes a central problem in computer vision and pattern recognition with a huge range of potential applications [21,26,27], including the fields of surveil-

[*] Corresponding author.
E-mail addresses: cpanag@ics.forth.gr (C. Panagiotakis), papoutsa@ics.forth.gr (K. Papoutsakis), argyros@ics.forth.gr (A. Argyros).

https://doi.org/10.1016/j.patcog.2018.02.001
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

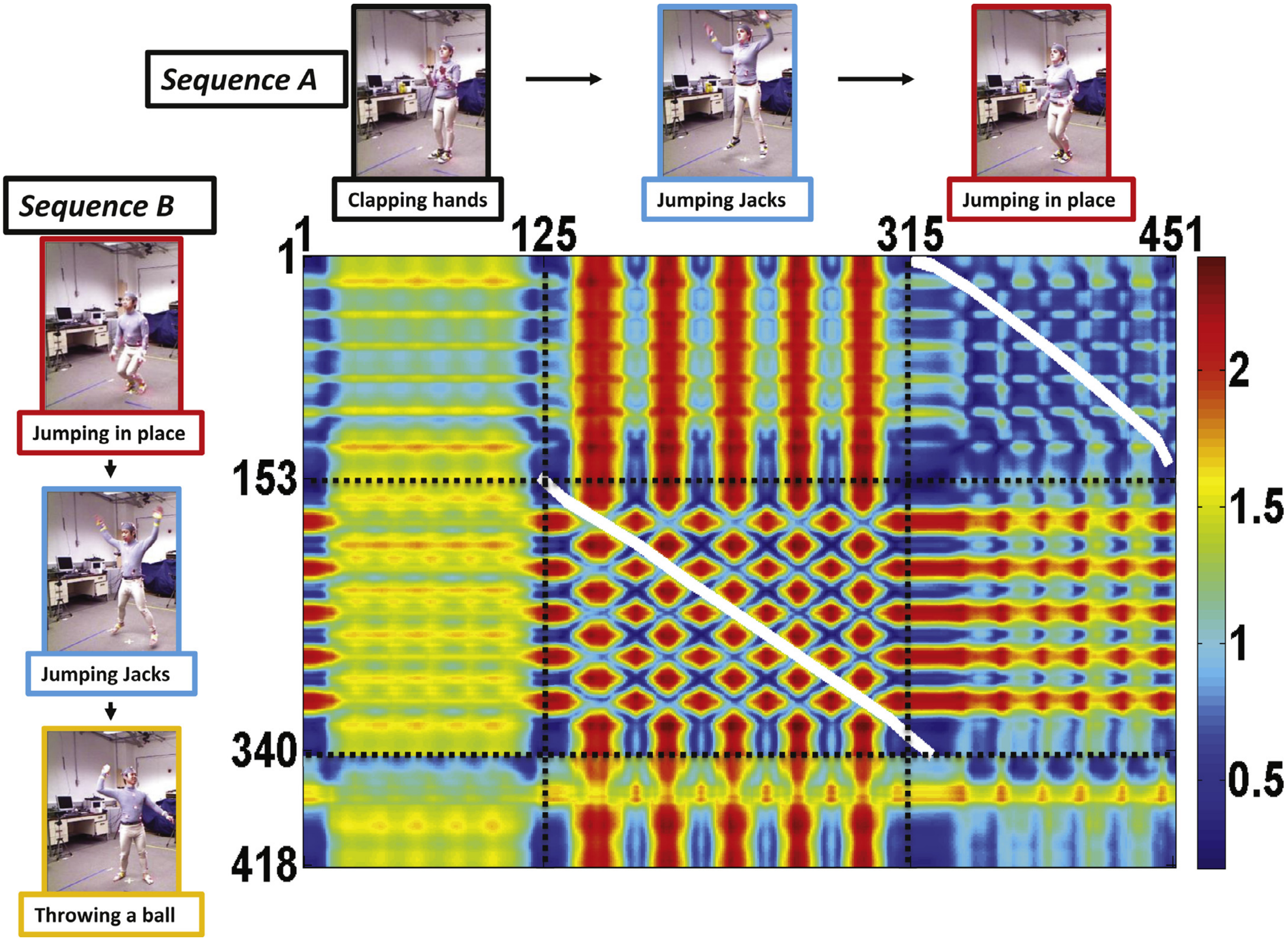


Fig. 1. Two detected commonalities (white curves) projected on the corresponding distance matrix, which was computed based on the pair-wise Euclidean distances between the frames of the image sequences A and B.

lance, advanced human computer interaction, content-based video retrieval, abnormal or suspicious activities detection, health monitoring and athletic performance analysis.

In this work, we consider commonalities between two multidimensional time series A and B, representing video or motion capture data. In such a setting, a candidate commonality is a pair of subsequences of A and B which can also be viewed as a path on the distance matrix D of all pairwise distances between the elements (frames) of A and B. Fig. 1 visualizes such a matrix in the form of a heat map, where two commonalities (white curves) are projected. Warm (cold) colors represent large (small) pairwise distances, respectively. The total cost of a path reflects how dissimilar the subsequences of a candidate commonality are. Low (high) cost paths correspond to similar (dissimilar) sub-sequences. A path of small length will tend to have low cost. However, it corresponds to a commonality of short subsequences and is probably not that interesting. As paths increase in length, their cost also increases. Thus, the trade-off between the length of the path (the duration of the commonality) and its cost should be balanced. Detecting multiple commonalities amounts to finding all paths in D that correspond to really common actions in A and B. The lack of supervision in this process has a twofold meaning: (a) no prior model or information on the actions is assumed to be known and (b) the number of commonalities is not assumed to be known a priori.

Given the potential commonality of two subsequences s_A and s_B of two sequences A and B, the corresponding commonality path and its associated cost can be estimated by employing Dynamic Time Warping (DTW) [28]. DTW is a widely-used algorithm for the optimal, non-linear temporal alignment of two sequences and has been extensively used for the alignment of time series [29] like human motion sequences [30] and speech/audio signals [31]. Recently, DTW has been successfully combined with canonical correlation analysis for temporal alignment of multi-modal data, such as acoustic and visual information [32]. The DTW cost is linear to the product of the lengths of the compared sequences, that is, $O(|s_A||s_B|)$. Thus, the naive approach to solve the multiple commonalities discovery problem would be to enumerate all possible paths, evaluate them and keep the best ones. Since the number of comparisons to be performed is $O(|A|^2|B|^2)$, it turns out that this exhaustive scheme has a complexity of $O(|A|^3|B|^3)$. This is prohibitive even for input sequences with a handful of frames. To deal with this problem, the proposed algorithm takes advantage of the properties of the distance matrix D and achieves state of the art performance with a computational complexity of $O(|A||B|)$. This makes possible the discovery of multiple commonalities even for input sequences of many thousands of frames.

In summary, the main contributions of this paper is (a) the formulation of the problem of unsupervised discovery of multiple commonalities in two time series as a search problem on a graph defined on the matrix of their pairwise frame distances and (b) the use of an efficient graph-based search algorithm for solving the problem. The proposed, deterministic solution requires no a priori knowledge on the number of commonalities, the durations of the matching subsequences or the types of actions. Moreover, the method is accurate and efficient. Specifically, in a series of experiments it is shown that, compared to the state of the art, the overlap of the unsupervised solutions to the ground truth increases by 6% on average and the required computational time is reduced by at least 50%.

2. Related work

In this section, we review relevant problems and solutions as they have been approached in different research communities.

Discovering motifs in time-series: Several methods in the data mining and knowledge discovery community deal with the problem of finding one or multiple common temporal patterns (motifs) within a single sequence [5]. A solution to this problem is useful in several domains ranging from biology and bio-informatics to computer science and engineering. Mueen and Keogh [5] proposed a method that discovers several motifs of different lengths. In [33], motif discovery is formulated as an optimization problem that is solved based on Particle Swarm Optimization. Moreover, Dynamic Time Warping (DTW) is used to define the objective function of the optimization problem, i.e., to quantify the similarity of different segments. In [34], Shou et al. introduce a multi-step processing technique for similarity search of query subsequences using DTW in multivariate time series. Their method decomposes each data sequence into a number of segments using a dimensionality reduction technique, and then applies a version of DTW on the segmented approximations of the data and query sequences to compute tight lower bounds for their DTW distance. The SwiftMotif method proposed in [8] segments a given time series with a data stream segmentation method and performs clustering based on segments similarity, where motifs may be defined. The fast time series segmentation and modeling techniques that are employed allow for an on-line detection of previously defined motifs in new time series and make SwiftMotif suitable for real-time applications. In [35], Emonet et al. proposed a model for unsupervised motif discovery that handles multivariate time series obtained from a mixture of activities coming from multiple modalities (videos from static cameras and audio localization data). Their approach is based on non parametric Bayesian methods to describe both the motifs and their occurrences in documents. In [6], Vahdatpour et al. address the problem of unsupervised activity and event discovery as multi-dimensional motif discovery in time series. First, their method extracts all single dimensional motifs. In the second stage, all single dimensional motifs are used to build a coincidence graph that is based on the temporal coincidence of those motifs in different time series dimensions. A graph clustering approach is then proposed to construct activity primitives. The work of Minnen et al. [36] also proposed an unsupervised approach for activity discovery in sensor data. It extended the work of Chiu et al. [37] on finding approximately repeated subsequences in single dimensional time series. It enables motif discovery in multidimensional sensory data and the automatic estimation of the size of each motif neighborhood, that is a crucial user-specific parameter for efficient detection of motif occurrences, highly dependent on the domain and the distance metric used to measure subsequence similarity, improving overall accuracy and quality of motif discovery.

Discovering common patterns in speech, images, videos: The dynamic programming algorithm presented in [38] is a segmental variant of Dynamic Time Warping. This method discovers and segments in an unsupervised manner all pairs of similar subsequences within two sequences, by exploiting the structure of repeating patterns within the speech signal. Subsequently, the method builds an inventory of lexical speech units that constitute the most representative ones within the given sequences. In image analysis, the term *co-segmentation* was introduced in [39] to define the task of jointly segmenting "something similar" given a set of images. This may refer to one or more objects of interest [40], or to a prominent image region [41] shared among some or all of the given images. The same method can be applied to a single image [42] to discover repeating spatial patterns. The same idea was extended to video segmentation [43] or to perform fore-/background video co-segmentation or single object co-segmentation in videos [44]. Chiu et al. [45] proposed a method to perform multi-class video object co-segmentation, in which the number of object classes and the number of instances are unknown in each frame and video. However, strong assumptions regarding the presence of objects or regions of interest in all frames from all videos are imposed. The work presented by Wang et al. [46] relaxes this assumption and is able to apply multiple video-based object co-segmentation for multiple videos, in which the target object may not be present in all frames.

Discovering common action patterns: Motivated by the success of methods in the previous tasks, several methods have been proposed for the discovery of common action-related patterns in motion capture data and videos. The method in [3] performs common action extraction in a pair of videos by segmenting the frames of both videos that contain the common action. To achieve this, the method relies on measuring the co-saliency of dense trajectories of spatio-temporal features. The method proposed by Zhou et al. [10] discovers facial units in video sequences of one or more persons in an unsupervised manner. The method relies on temporal segmentation and clustering of sequences containing facial features. In a more recent work, Zhang and Mahoor [47] proposed a method for simultaneous detection of multiple facial action units (AUs) based on their co-occurrence relationships in human facial activities (emotions). In this approach, the detection of each AU is considered as a task. Discovering all AUs simultaneously is modeled as a multi-task multiple kernel learning (TD-MTMKL) problem that optimizes a trade-off between capturing commonalities and adapting to variations in modeling of AU inter-relations.

Another recently work by Yeo et al. [48] introduces an unsupervised learning algorithm to detect a common activity (co-activity) from a set of videos, which is formulated using absorbing Markov chain. The method detects a common activity (co-activity) of variable length in two or more videos or identifies multiple instances of a co-activity in a single video. Chu et al. [13] propose a video co-summarization technique which can be applied to the co-activity detection problem. They introduce the Maximal Biclique Finding (MBF) algorithm operating on complete bipartite subgraphs among frames of two paired videos to determine sparsely co-occurring spatio-temporal patterns. Their method is also extended to multiple videos by aggregating pairwise results.

An interesting formulation for discovering common events in an unsupervised manner is presented by Yuan et al. [49] and noted as a task of recurring event mining. Recurrent events are defined as short temporal patterns that consist of multiple instances in a target database. This task is translated into finding temporally continuous paths in a matching trellis simulated by a "forest-growing" procedure, where each path indicates a repetition of an event. The method was applied to video or motion capture human motion data and was robust under large temporal and content variations of the repetitions of the common patterns. Given as input an online video stream capturing a scene in which the same action is repeated multiple times in consecutive cycles, the method of Levy et al. [50] is able to detect the start and end points of the sequence of repetitive actions, and counts the repetitions. The work of Shariat et al. [51] combines the discovery of common action patterns and action classification, introducing an adaptive segmental alignment model that is able to detect the boundaries of temporal segments representing common actions and efficiently matching them.

One of the most related methods to the one proposed in this paper is the method by Chu et al. [1] that discovers multiple common actions in a pair of videos or time-series. The problem is noted as Temporal Commonality Discovery (TCD). It is treated as an integer optimization problem by proposing the branch-and-bound (B& B) algorithm [52] for efficient searching simultaneously over all possible segments in each video sequence, modeled as histograms that are compared using the χ^2 distance. The method is generic and can be applied to any histogram-based feature. Our method is also generic, without requiring histogram-based features. Additionally, our method is fully unsupervised since it is able to automatically determine the number of commonalities, while TCD requires the number of commonalities to be a priori known. Another recently proposed method [4] treats the multiple commonality discovery problem as a stochastic optimization problem solved by employing Particle Swarm Optimization with an objective function defined based on the non-linear DTW alignment cost of two sub-sequences. Two variants were proposed, one that assumes that the number of commonalities is known (*S-EVACO*) and one that does not require that information (*U-EVACO*). In [4] it has been shown that the EVACO variants clearly outperform the other state of the art methods. In this paper, we show that the deterministic methods we propose lead to better results in less computational time.

The rest of the paper is organized as follows: Section 3 sets the scene by presenting the various aspects of the problem, analyses the properties of the problem and presents ideas that are used to cope with the computational complexity of the problem. Section 4 capitalizes on this formulation and findings to present the proposed algorithmic solutions. The experimental results and comparisons with existing methods are given in Section 5. Finally, conclusions and discussion are provided in Section 6.

3. Problem constraints and formulation

We assume two input sequences A and B of lengths $|A|$ and $|B|$, respectively, and the $|A| \times |B|$ matrix D of the pair-wise distances of their frames (see Fig. 1). Depending on the nature of the sequences, different frame representations and distance functions can be employed. A commonality is represented as a connected path of points $(x_i,\ y_i)$ for which it holds that $\forall x_i, y_i,\ \ x_i \leq x_{i+1},\ \ y_i \leq y_{i+1}$. Besides this constraint, paths can start and end anywhere in D. In our formulation, a subsequence q_A of sequence A is represented as $q_A = [s_A, e_A],\ e_A \geq s_A$, where s_A, e_A are the start and end frames of the subsequence, respectively. A commonality $c = \langle q_A, q_B \rangle = \langle [s_A, e_A], [s_B, e_B] \rangle$ of A and B is a pair of subsequences q_A (of A) and q_B (of B) that represent the same action. Fig. 1 gives an example of a particular distance matrix D obtained after comparing the frames of two sequences. Two commonalities, $c_1 = \langle [125, 315], [153,\ 340] \rangle$ and $c_2 = \langle [315, 451], [1,\ 153] \rangle$ are illustrated. A commonality $c = \langle q_A, q_B \rangle = \langle [s_A, e_A], [s_B, e_B] \rangle$ defines the rectangle b_c on D, with $(s_A,\ s_B)$ being the top left and $(e_A,\ e_B)$ the bottom right corner of the rectangle. The actual correspondence between frames of the subsequences q_A and q_B of a commonality are determined by the minimum cost path in D connecting $(s_A,\ s_B)$ with $(e_A,\ e_B)$. Essentially, b_c is the bounding box of this path. Table 1 summarizes the notation used throughout this work.

As stated in Section 1, the computational cost of the exhaustive method for finding a single commonality is $O(|A|^3|B|^3)$. This is too costly even for input sequences of only a few decades of frames. We capitalize on the properties and the structure of the problem to propose an algorithm that discovers all commonalities of two sequences consisting of thousands of frames.

3.1. Commonality endpoints

Let $c = \langle q_A, q_B \rangle = \langle [s_A, e_A], [s_B, e_B] \rangle$ be a candidate commonality. It is reasonable to assume that a commonality is not expected to start (or end) at a pair of frames that are quite dissimilar. This means that both $D(s_A,\ s_B)$ and $D(e_A,\ e_B)$ should be lower than a threshold T_L. To exploit this, we first define the set $\mathcal{L}$ of points p that constitute local minima of the distance matrix D. Then, we define the subset $\mathcal{E}$ of $\mathcal{L}$ as

$$\mathcal{E} = \{p \in \mathcal{L} : D(p) < T_L\}. \tag{1}$$

The set $\mathcal{E}$ contains the local minima of matrix D whose value is lower than a threshold T_L. T_L is automatically determined by an unsupervised statistical analysis method based on the properties of the distribution of the values of the local minima of a distance matrix D, that is, the D values of points $p \in \mathcal{L}$. More specifically, let $f_{\mathcal{L}}(p)$ be the cumulative distribution of this function. Then, T_L is the value with $f_{\mathcal{L}}$ equal to 0.5.

The two endpoints of a commonality are restricted to belong to $\mathcal{E}$.

3.2. Commonality midpoints

We restrict commonality paths to be polygonal lines that pass from suitably identified points in D. Similarly to commonality endpoints, we expect the midpoints of a commonality to be local minima of D. Thus, we restrict the set $\mathcal{M}$ of commonality midpoints to be a subset of $\mathcal{L}$. In notation,

$$\mathcal{M} = \{p \in \mathcal{L} : D(p) < T_H\}, \tag{2}$$

where T_H is a high threshold $(T_H > T_L)$ that is automatically determined similarly to T_L. T_H is the value with $f_{\mathcal{L}}$ equal to 0.9. T_H and T_L have been set experimentally and were kept constant for all experiments and datasets. Thus, the setting of the T_L and T_H thresholds individually per dataset is avoided. Given that $T_H > T_L$, it turns out that $\mathcal{E} \subseteq \mathcal{M}$.

Additional points p are iteratively included in $\mathcal{M}$ in order to improve the accuracy of the polygonal line approximation of a commonality path. To achieve this, we keep including to $\mathcal{M}$ points p with the lowest possible value in D, under the constraints that (a) $D(p) < T_H$ and (b) there is no point in $\mathcal{M}$ whose Euclidean distance from any point in $\mathcal{M}$ is shorter than $T_D = 15$ points. Fig. 2 shows the sets $\mathcal{E}$ and $\mathcal{M}$ superimposed on the corresponding distance matrix of Fig. 1. In this example, $\mathcal{E}$ and $\mathcal{M}$ sets consist of 495 and 1070 points (including 219 extra points), respectively.

3.3. Subsequences length and commonalities scale

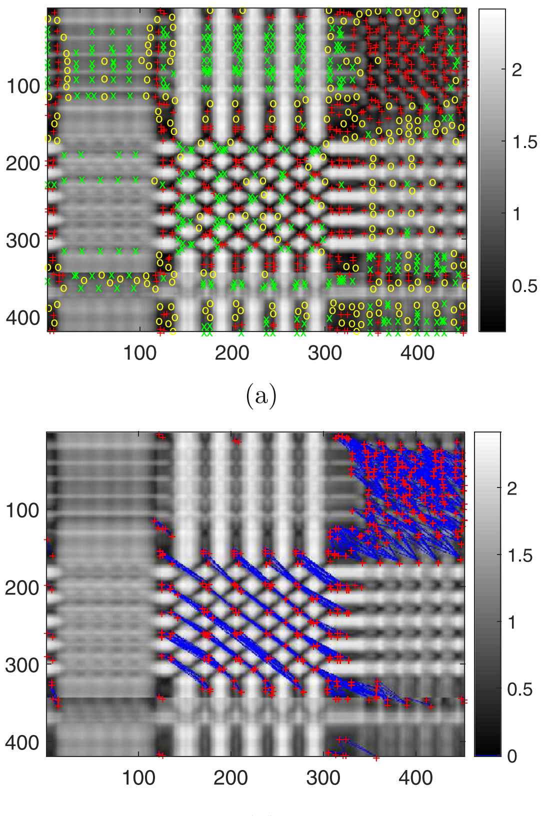
We consider that sequences that are shorter than a minimum, dataset-dependent length do not constitute meaningful actions, so we enforce this size constraint also to potential commonalities. Moreover, in several situations, it is quite unnatural to match two subsequences of quite different lengths. We define the scale σ_c of a commonality $c = \langle [s_A, e_A], [s_B, e_B] \rangle$ to be

$$\sigma_c = \frac{max(e_A - s_A, e_B - s_B)}{min(e_A - s_A, e_B - s_B)}. \tag{3}$$

All commonalities c with $\sigma_c > \sigma_{max} = 5$ are rejected. This constraint is used to filter unlikely commonalities and, thus, speeds up the method. In scenarios where commonalities differ considerably in length, the constraint can be relaxed by setting σ_c to a higher value.

Table 1
Summary of the notation used throughout this work.

Symbols	Definitions
A, B	Two given sequences of lengths $\lvert A\rvert$, $\lvert B\rvert$
D	Distance matrix of the pair-wise distances for all frames of A, B
$q_A = [s_A, e_A]$	A subsequence of A, where s_A, e_A are the start and end frames
$c = \langle q_A, q_B\rangle$	A commonality is a pair of subsequences q_A and q_B
$A(c)$	The area of the commonality rectangle
$P(c)$	The cost of the commonality c (e.g. DTW cost)
$\omega(c)$	Objective function for the single commonality selection problem
$\Omega(C)$	Objective function for the multiple commonality selection problem



(a)

(b)

Fig. 2. (a) A sample distance matrix and the associated sets of commonality end- and mid-points. The "+" symbol marks points that belong to $\mathcal{E}$. The "$\times$" symbol marks points that belong to $\mathcal{M}$ but not to $\mathcal{E}$. Finally, the "o" symbol marks the extra commonality midpoints added to $\mathcal{M}$. (b) The nodes (red "+") and the edges (blue lines) of the subgraph G' (see text for details). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.4. Graph modeling

We construct a directed graph $G = (V, E)$. The set V of nodes of this graph are the points of the set $\mathcal{M}$. Assume two nodes $u = (s_i, s_j)$ and $v = (e_i, e_j)$, $e_i > s_i$, $e_j > s_j$, u, $v \in V$. These nodes are connected with a directed edge $e = (u, v)$ if and only if:

1. $|v - u|_\infty \leq 2 \cdot T_D$. This means that v is inside a square of size equal to $2 \cdot T_D$ whose top-left corner is located on u.
2. For all points p of the matrix D that are under the straight line connecting u and v it holds that $D(p) < T_H$.

Intuitively, the above two constraints guarantee that the straight line connecting u and v can be a valid segment of a commonality, i.e., the subsequence $[s_i, e_i]$ is similar to the subsequence $[s_j, e_j]$.

Each edge $e = (u, v) \in E$ is associated with a weight

$$d(u,v) = \left(2 - \frac{1}{\sigma_c}\right) \sum_{p \in |uv|} D(p). \tag{4}$$

The cost $d(u, v)$ represents the sum of the values of the distance matrix D under the straight line connecting u and v, weighted by a factor depending on the scale σ_c of the commonality $c = \langle [s_i, e_i], [s_j, e_j]\rangle$ corresponding to the bounding box b_c of (u, v). Intuitively, this weighting scheme gives a smaller cost to commonality segments of $\sigma_c = 1$, that is, common subsequences of equal length.

3.5. Finding all shortest paths

After its construction, the graph $G = (V, E)$ is simplified to its subgraph $G' = (V', E')$. Specifically, the set $V = \mathcal{M}$ is reduced to the set $V' = \mathcal{E}$ by removing from V the nodes $\mathcal{M} - \mathcal{E}$. At the same time, the removal of these nodes results in the removal of edges in E. When midpoints are removed from G, the weights of the remaining edges are updated properly so that graph connectivity and shortest path costs of G are maintained in G. This is done in a way that guarantees that for every pair of nodes $u, v \in V' = \mathcal{E}$, if those were connected with a path of minimum cost λ in G, they will stay connected in G' with a minimum cost path of λ, too. Thus, the solutions provided by our method on G are guaranteed to be exactly the same with those on G.

We apply Johnson's algorithm [53] to find the shortest paths between all pairs of vertices in the edge weighted graph $G' = (V', E')$. Johnson's algorithm has a time complexity of $O(|V'| \cdot \log(|V'|) + |V'| \cdot |E'|)$, where $|V'|$ and $|E'|$ are the number of nodes and edges of the graph, respectively. Removing $M - E$ only aims for computational efficiency and means that Johnsons algorithm operates only on candidate endpoints, avoiding midpoints.

3.6. Evaluating a single commonality

Solving the single commonality discovery problem amounts to finding the commonality c^* that maximizes an appropriately defined objective function ω. In notation,

$$c^* = argmax_c \quad \omega(c). \tag{5}$$

Two subsequences $q_A = [s_A, e_A]$, and $q_B = [s_B, e_B]$ define the candidate commonality $c = \langle q_A, q_B\rangle$. In order to assess this commonality, we propose an objective function ω that consists of two terms:

- The cost $P(c)$ of the commonality c, defined as the cost of the minimum path from node $u = (s_A, s_B)$ to node $v = (e_A, e_B)$. This is estimated by the invocation of the Johnson's algorithm (Section 3.5) on G'. The cost $P(c)$ is equal to the sum of $d(i, j)$ along the identified minimum path for c. Intuitively, the larger this cost is, the less preferable the commonality c.
- The product $A(c) = |q_A||q_B| = (e_A - s_A) \cdot (e_B - s_B)$ of the lengths of the two sub-sequences. Intuitively, the objective function ω should favor the matching of larger sub-sequences. The term

$A(c)$ is equal to the area of the bounding rectangle b_c of the corresponding commonality path in D.

There is a trade-off between the terms $A(c)$ and $P(c)$. Commonalities c with large $A(c)$ are preferable. At the same time, as $A(c)$ increases, $P(c)$ also increases. This trade-off is captured by defining the objective function $\omega(c)$ as

$$\omega(c) = \frac{A(c)}{P(c) + \epsilon}. \tag{6}$$

Theoretically, $P(c)$ might be equal to zero. Thus, ϵ is a small constant preventing division by zero. In our work, ϵ was set equal to the 1% of the global minimum non-zero entry of the distance matrix D.

3.7. Evaluating multiple commonalities

We are interested in discovering multiple commonalities between two sequences without a priori knowledge of their number. The solution to the multiple commonalities discovery problem is a set of commonalities $C = \{c_1, c_2, \ldots, c_{|C|}\}$ that maximize a suitable objective function $\Omega(\cdot)$, i.e.,

$$C^* = argmax_C \quad \Omega(C). \tag{7}$$

The proposed objective function $\Omega(C)$ is defined as:

$$\Omega(C) = \frac{A(C)}{\sum_{i=1}^{|C|} P(c_i) + \epsilon}. \tag{8}$$

In Eq. (8), the enumerator $A(C)$ is a generalization (for all commonalities in C) of the term $A(c)$ defined for a single commonality. Let X be the subset of all frames of A that are members of some commonality $c_i \in C$. Similarly, let Y be the subset of all frames of B that are members of some commonality $c_i \in C$. Then, $A(C) = |X| \cdot |Y|$. Intuitively, this definition considers the single, “super-commonality” involving all frames of the two sequences A and B and estimates its area (as in the case of Eq. (6)). A nice property of this definition is that it is conceptually compatible to the one defined for the case of a single commonality. More specifically, assume that we view a single commonality c as two consecutive, non overlapping commonalities c_1 and c_2. Then, the evaluation of c_1 and c_2 in Eq. (8) gives the same score as the evaluation of c in Eq. (6).

4. The MUCOS and SMUCOS algorithms

Based on the problem formulation and constraints presented in Section 3, we now present the proposed algorithms for solving the problem of discovering multiple commonalities in two sequences.

The MUCOS algorithm: *MUCOS* solves the *MUltiple COmmonalitieS* discovery problem. To discover all commonalities of two sequences *A, B, MUCOS* operates as follows:

1. Compare pairwise all frames of the two sequences *A, B* to come up with their distance matrix *D*.
2. Compute the sets $\mathcal{L}$, $\mathcal{E}$ (Section 3.1) and $\mathcal{M}$ (Section 3.2).
3. Define the graph $G = (V, E)$ and its sub-graph $G' = (V', E')$ (Section 3.4).
4. Compute all shortest paths in G' (Section 3.5).
5. Each shortest path resulting from the previous step is associated with a commonality. Discard commonalities that do not meet the criteria for the length of the subsequences and the scale of the commonality (Section 3.3). Let the remaining candidate commonalities be the set $S = \{c_1, c_2, \ldots, c_{|S|}\}$.
6. Start with an empty solution set $C = \emptyset$ of commonalities and its score $\Omega(C) = 0$. Check which commonality $c \in S$ maximizes $\Omega(C)$ as defined in Eq. (8). If there is such a commonality, introduce it in C and remove it from S. Otherwise, terminate.

An important property of graph G' is that, typically, consists of weakly connected components, each associated with a single commonality. Therefore, the execution of the Johnson’s algorithm and the optimization of the objective function can be performed independently in each connected component, achieving the decomposition of the whole problem into several, simpler ones.

The SMUCOS algorithm: In the case that the number of commonalities to be detected is known/given, we can modify the step (6) of *MUCOS* to terminate the algorithm when the number of the selected commonalities is equal to the given number. We denote this variant of the algorithm as *SMUCOS* which stands for *Supervised MUltiple CommonalitieS discovery*.

4.1. Scalability

The proposed method requires the computation of the pairwise distances between the frames of the two input sequences. Thus, its direct use for discovering commonalities in very large input sequences (e.g., sequences of tens of million frames as in [54]) is problematic. However, with a straightforward decomposition of the problem, it is still possible to handle sequences with length in the order of millions.

More specifically, this can be achieved by splitting the largest of the two input sequences into a number of non overlapping segments of equal length. Then, a set of distance matrices is computed between each segment of the largest sequence and the smallest sequence. The proposed method can be executed for each of the resulting distance matrices. As a final step, the detected commonalities are merged. Consider, for example, two input sequences with 1M and 2M frames, respectively. We split the largest one (2M) into 2000 sequences of 1K, resulting in 2000 distance matrices of dimensions 1M × 1K. Each of them has the manageable size of 4GB when the matrix values are stored as float numbers.

5. Experimental evaluation

We assess experimentally several aspects of the performance of *MUCOS* and *SMUCOS* by comparing with the following state of the art methods: *S-EVACO* and *U-EVACO* [4], *TCD* [1], the method proposed by Guo et al. [3] for video co-segmentation and the Segmental DTW (*SDTW*) [38]. The code implementing the proposed method together with experimental results are publicly available online.[1]

5.1. Datasets and performance metrics

The experimental evaluation was conducted using the four datasets[2] presented in [4], consisting of 373 pairs of sequences, 2355 action sub-sequences and 1286 pairs of common actions. More specifically:

- **MHAD101-s dataset**: 101 pairs of action sequences of skeletal data. Each sequence consists of 3–7 actions and each pair has 1–4 common actions. Sequences were defined based on the Berkeley Multimodal Human Action Database (MHAD) [55] that contains human motion capture data as well as conventional RGB video and depth data. The human pose is represented as a $30 + 30 + 4 = 64$D vector. The first 30 dimensions encode angles of selected body parts with respect to a body-centered coordinate system. The next 30 dimensions encode the same angles but in a camera-centered coordinate system. Finally, this representation is augmented with the four angles between the 3D vectors of the fore- and the back-arms as well as the angles between the upper- and lower legs [4].

[1] https://sites.google.com/site/costaspanagiotakis/research/mucos.

[2] Available at http://www.ics.forth.gr/cvrl/evaco/.

Table 2
Evaluation results on the **MHAD101-s** dataset.

Methods	$\mathcal{R}$(%)	$\mathcal{P}$(%)	F_1(%)	$\mathcal{O}$(%)
U-SDTW [38]	65.8	45.5	47.7	35.1
U-EVACO [4]	71.3	63.9	63.3	50.3
MUCOS	**86.0**	**69.4**	**74.9**	**64.6**
TCD [1]	16.7	18.1	13.8	8.5
S-SDTW [38]	61.6	47.1	48.5	35.9
S-EVACO [4]	77.9	67.6	71.3	59.4
SMUCOS	**82.4**	**77.4**	**78.7**	**69.9**

Table 3
Evaluation results on the **CMU86-91** dataset.

Methods	$\mathcal{R}$(%)	$\mathcal{P}$(%)	F_1(%)	$\mathcal{O}$(%)
U-SDTW [38]	44.9	20.9	27.6	16.1
U-EVACO [4]	**71.3**	**67.4**	**65.2**	**51.0**
MUCOS	63.4	59.8	57.9	43.0
TCD [1]	30.9	51.3	38.0	24.1
S-SDTW [38]	44.9	20.9	27.6	16.1
S-EVACO [4]	**67.6**	**77.1**	**71.6**	**57.5**
SMUCOS	66.2	69.9	67.1	53.0

Table 4
Evaluation results on the **MHAD101-v** dataset.

Methods	$\mathcal{R}$(%)	$\mathcal{P}$(%)	F_1(%)	$\mathcal{O}$(%)
U-SDTW [38]	69.4	45.7	48.0	35.5
U-EVACO [4]	63.3	**63.3**	**58.8**	**45.9**
MUCOS	**83.0**	50.4	54.3	41.2
TCD [1]	20.6	14.0	15.4	19.3
S-SDTW [38]	65.2	49.1	50.5	37.7
S-EVACO [4]	76.6	66.8	69.8	56.2
SMUCOS	**78.6**	**72.1**	**72.1**	**59.7**

Table 5
Evaluation results on the **80-Pair** dataset.

Methods	$\mathcal{R}$(%)	$\mathcal{P}$(%)	F_1(%)	$\mathcal{O}$(%)
U-SDTW [38]	34.6	60.6	37.3	25.6
Guo [3]	55.6	**78.1**	60.9	51.6
U-EVACO [4]	61.0	69.7	62.0	54.2
MUCOS	**87.2**	72.7	**73.9**	**64.0**
TCD [1]	22.9	65.4	31.2	21.5
S-SDTW [38]	27.8	52.2	31.4	21.6
S-EVACO [4]	75.8	77.2	73.9	**64.5**
SMUCOS	**78.8**	**78.0**	**74.3**	63.3

- **CMU86-91 dataset**: Contains 91 pairs of action sequences of skeletal data. Pairs include combinations of 14 long action sequences of the set Subject 86 of the CMU-Mocap database. Each action sequence consists of up to 10 actions executed in a continuous manner. The feature representation of human motion data is based on the position and orientation of the skeletal root and relative joint angles that results in a 30-D feature vector per frame [4].
- **MHAD101-v dataset**: The MHAD101-v dataset is identical to MHAD101-s in terms of action composition and ground truth, but uses the RGB video stream instead of the motion capture data. The representation is based on the Improved Dense Trajectories (IDT) features [4]. Four types of descriptors, namely trajectory shape, HOG, HOF, and MBH are encoded by a Bag-of-Features representation, separately for each type of descriptor and for each pair of videos in the dataset [4].
- **80-Pair dataset**: The 80 pairs of the dataset consists of 50 segmented clips of human actions from the UCF50 dataset [56] and 30 pairs selected from BBC animal documentaries depicting animal actions [3]. Each frame is represented by a 25D feature vector that is the histogram of frequencies of the codewords for the trajectories ending up in that frame. The 25 codeword are defined the application of the k-means method on a Bag-of-Features representation based on the MBH descriptors of all frames for a pair of videos [4].

These datasets involve time series of skeletal data (MHAD101-s, CMU86-91 datasets) as well as real RGB videos (MHAD101-v, 80-Pair datasets). The datasets provide access to the raw data but also to the features representing each frame permitting the comparison of frames. For the fairness of the comparison to existing methods, we used exactly the same frame representations and comparisons proposed in [4].

In order to assess the performance of the evaluated methods, we employed the established metrics of precision, recall, F_1 score and overlap (intersection-over-union), as reported in [4]. Precision quantifies how many of the frames of the co-segmented sequences belong to the set of commonalities in both sequences. Recall quantifies how many of the actual commonalities (common frames) are indeed discovered/segmented by the method.

5.2. Comparisons with state of the art methods

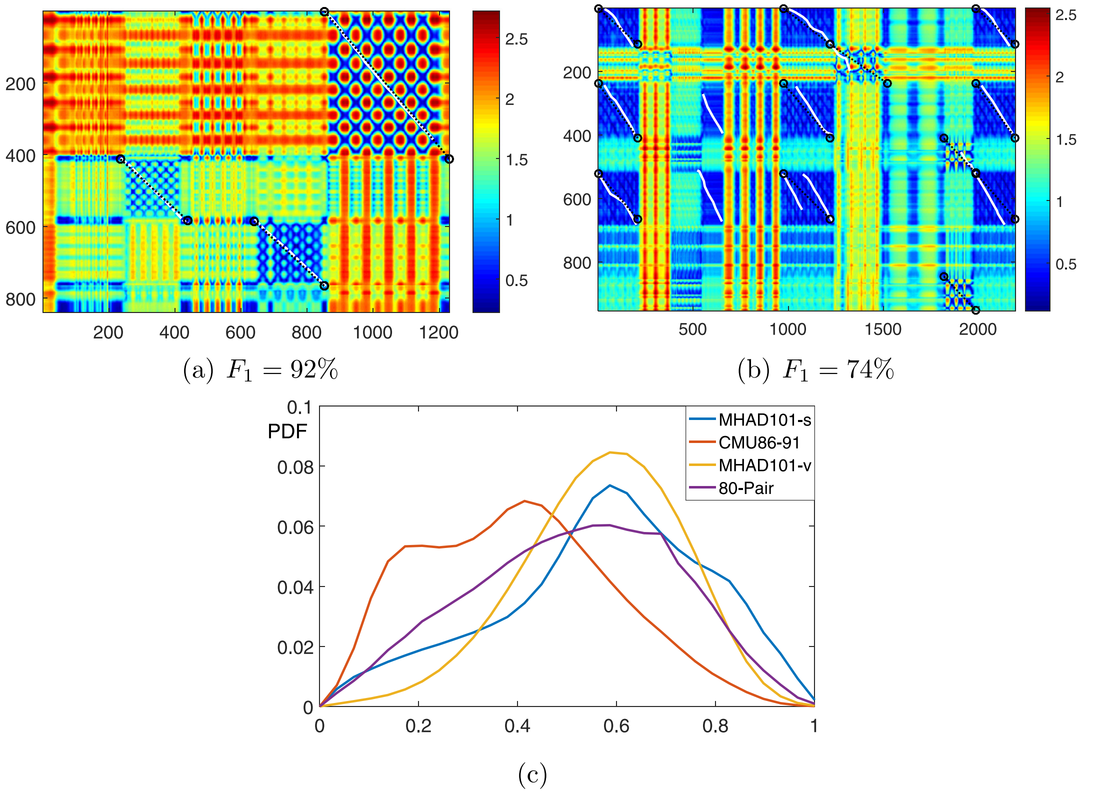
Tables 2–5 show the results obtained on the MHAD101-s, CMU86-91, MHAD101-v and 80-Pair, respectively. The scores are presented as % average scores computed over all individual scores per sample (pairs of sequences) of a dataset. We report the performance of all evaluated methods on all aforementioned metrics. The results for the existing methods are those reported in [4] and are copied here for convenience. Each table is split in two parts, the top rows that report results of unsupervised methods (*U-SDTW, U-EVACO, Guo* [3] and *MUCOS*), i.e., the ones where the number of commonalities is not known a priori. The rest of the rows report results of supervised methods (*TCD, S-SDTW, S-EVACO, SMUCOS*), that is, methods that require knowledge of the number of commonalities.

MUCOS outperforms all the corresponding unsupervised state of the art methods on two out of four datasets (MHAD101-s, and 80-Pair dataset) and has the second highest performance on the CMU86-91 and MHAD101-v datasets. *SMUCOS* outperform all the corresponding supervised state of the art methods on three out of four datasets (MHAD101-s, MHAD101-v and 80-Pair dataset) and has the second highest performance on the CMU86-91 dataset. The *EVACO* variants outperform all the corresponding unsupervised and supervised state-of-the-arts methods on CMU86-91 dataset. *Segmental DTW* outperforms *TCD* on three out of four datasets (MHAD101-s, MHAD101-v and 80-Pair dataset). Guo method [3] yields the third highest performance on 80-Pair dataset.

The best performance of the proposed variants is reported on MHAD101-s, where the average overlap of *MUCOS* and *SMUCOS* is 28% and 18% higher than the average overlap of *U-EVACO* and *S-EVACO*, respectively. The worst performance of *MUCOS* and *SMUCOS* is reported on CMU86-91, where the overlap of *U-EVACO* and *S-EVACO* is 18% and 9% higher than the overlap of *MUCOS* and *SMUCOS*, respectively.

In our effort to understand why *MUCOS* and *SMUCOS* do not have top performance in the CMU86-91 data set as it happens with the rest of the datasets, we investigated the skewness γ_1 [57] of the distributions of normalized values of the distance matrices D in each dataset, defined as

$$\gamma_1 = \frac{E[(d-\mu)^3]}{\sigma^3}, \tag{9}$$



(a) $F_1 = 92\%$ (b) $F_1 = 74\%$

(c)

Fig. 3. Results of MUCOS on a pair of **(a)** MHAD101-s and **(b)** CMU86-91 dataset. The selected commonalities (white curves) and the ground truth (black dotted curves) are projected on the corresponding distance matrix. **(c)** The PDFs of normalized values of distance matrices per dataset.

where μ is the mean, σ is the standard deviation and E is the expectation operator. Fig. 3 plots these distributions for the four datasets. The average skewness for CMU86-91 is 0.20 (positive), while the average skewness for the MHAD101-s, MHAD101-v and 80-pair is -0.43, -0.37 and -0.11 (negative), respectively. The positive skewness of CMU86-91 means that the mass of the distribution is concentrated to the left, explaining the existence of indistinguishable local minima that are used by the proposed method to identify and then evaluate candidate commonalities. As a concrete example, Fig. 3 depicts the selected commonalities (white curves) of *MUCOS* and the ground truth (black dotted curves) projected on the corresponding distance matrix D of a pair of (a) MHAD101-s and (b) CMU86-91 dataset. On the example from MHAD101-s, *MUCOS* gives a solution with F_1 score = 92%, while on the example from CMU86-91 the solution has F_1 score equal to 74%. In the example from the CMU86-91 dataset, the distance matrix is smoother without strong local minima. This is in contrast to the example from MHAD101-s.

Fig. 4 summarizes the findings in motion captured (top) and video (bottom) datasets. It shows the mean F_1 score for all sequence pairs, after zeroing the F_1 score of pairs below an overlap threshold on motion captured (Fig. 4(a)) and video (Fig. 4(b)) datasets. The proposed methods *MUCOS* and *SMUCOS* correspond to the black curves. The performance of supervised and unsupervised methods is illustrated as dotted and continuous lines, respectively. *MUCOS* outperforms *U-EVACO* on motion captured datasets and for high overlap threshold values on video based datasets. *U-EVACO* outperforms *MUCOS* for low overlap threshold values on video datasets. *SMUCOS* outperforms *S-EVACO* under any type of dataset and overlap threshold value. Overall, it can be observed that *MUCOS* and *SMUCOS* outperform or are in par with the top performing methods in all datasets.

By aggregating the obtained results over all datasets, it turns out that the proposed supervised and unsupervised variants of the method improve the overlap criterion by 4% and 6%, respectively, in comparison to the corresponding top performing existing methods [4] (*S-EVACO* and *U-EVACO*).

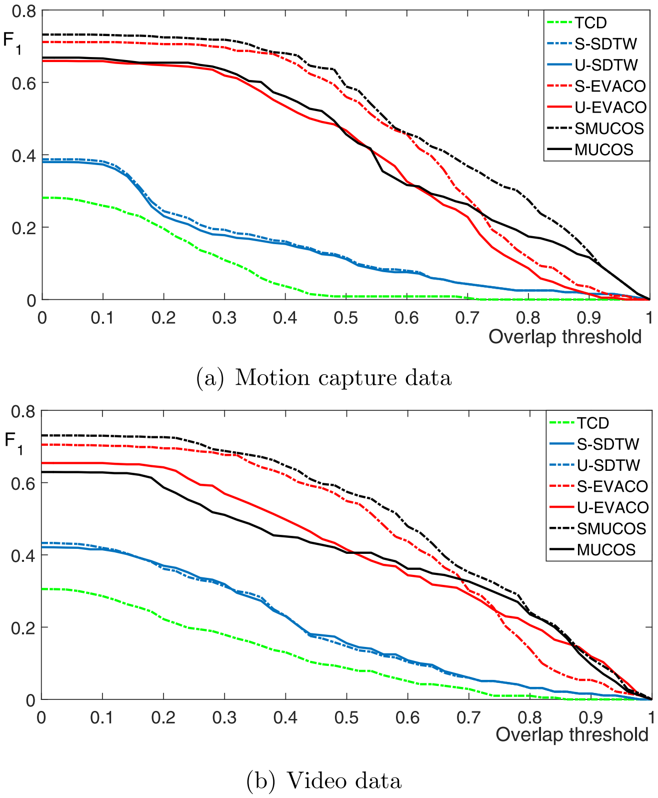
5.3. Computational performance

MUCOS and *SMUCOS* have been implemented using MATLAB. All experiments were executed on an Intel I7 CPU processor at 2.4 GHz. Typical processing times for the execution of *MUCOS* for $1k \times 1k$ and $10\ k \times 10\ k$ distance matrices are 4 sec and 5 minutes, respectively. The computational efficiency of our method is the result of:

- the approximation of commonality paths by polygonal lines connecting endpoints and midpoints, see Section 3.1 and 3.2.
- The fact that the graph of the problem consists of weakly connected components and that the Johnsons algorithm can be applied to each of them, individually (see Section 4). This means that if there is an upper bound on the lengths of common actions, the problem complexity increases linearly with the sizes $|A|$ and $|B|$ of the input sequences. Indeed, this is illustrated in the scatter plot of Fig. 5 where the execution times of *MUCOS* are plotted as a function of the number of points of the corresponding distance matrices.

Comparable computational costs are only achieved by Segmental DTW, which, nevertheless, provides solutions of much lower quality (see Figs. 4(a) and (b)).

We also computed the time required by *S-EVACO* and *SMUCOS* to process all datasets on a computer system with the same characteristics. We chose to compare *SMUCOS* with *S-EVACO* since the latter has been shown to be more efficient than the rest of the



(a) Motion capture data

(b) Video data

Fig. 4. Summary of the obtained results in all datasets.

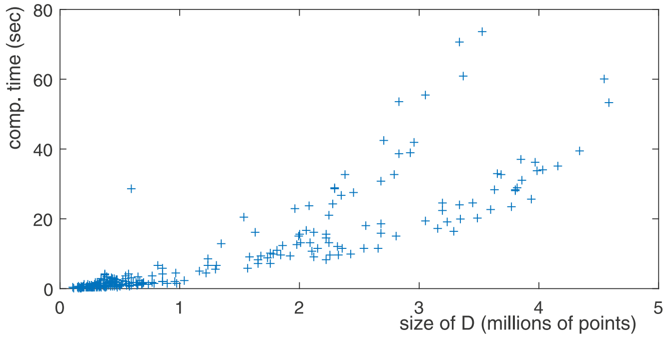


Fig. 5. The scatter plot of the execution times of *MUCOS* relative to the size of the associated distance matrices.

evaluated methods [4]. We investigate the supervised versions of the algorithms searching for the known number of commonalities in each pair. The comparison of the unsupervised versions would be in favor of *SMUCOS*, as the *U-EVACO* algorithm searches for an upper bound of commonalities and then selects the best out of them. Overall (sum of execution times in all datasets) *SMUCOS* is 50% faster than *S-EVACO*. This is despite the fact that *SMUCOS* is implemented in MATLAB, and *S-EVACO* is implemented in Python.

6. Summary

We proposed a method to solve efficiently the problem of discovering multiple common actions in time series and videos. Our approach discovers such commonalities without any prior knowledge on their type, number or duration. Our method outperforms the existing state of the art methods in all criteria and in most of the employed datasets. The quality of the solutions is combined with computational efficiency, as the proposed method is the fastest among the competing ones. Another advantage of *MUCOS* is its deterministic nature compared, e.g., to the up to now top-performing methods (*U-EVACO, S-EVACO* [4]) that are stochastic due to the Particle Swarm Optimization strategy they employ. Experiments on the challenging datasets of motion capture and video data proposed in [4] involve a variety of features and representations of time series and videos. This demonstrates that *MUCOS* and *SMUCOS* can be applied to a wide range of representations and, therefore, may constitute a useful tool in a broad range of applications.

Acknowledgments

This work was partially supported by the H2020 projects ACANTO and Co4Robots.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2018.02.001.

References

[1] W.-S. Chu, F. Zhou, F. De la Torre, Unsupervised temporal commonality discovery, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, 7575, Springer Berlin Heidelberg, 2012, pp. 373–387.

[2] C. Xiong, J.J. Corso, Coaction discovery: segmentation of common actions across multiple videos, in: Proceedings of the Twelfth International Workshop on Multimedia Data Mining, in: MDMKDD '12, ACM, New York, NY, USA, 2012, pp. 17–24, doi:10.1145/2343862.2343865.

[3] J. Guo, Z. Li, L.-F. Cheong, S.Z. Zhou, Video co-segmentation for meaningful action extraction, in: IEEE International Conference on Computer Vision, IEEE, 2013, pp. 2232–2239.

[4] K. Papoutsakis, C. Panagiotakis, A. Argyros, Temporal action co-segmentation in 3d motion capture data and videos, in: IEEE Conference on COmputer Vision and Pattern Recognition (CVPR), 2017.

[5] A. Mueen, N. Chavoshi, Enumeration of time series motifs of all lengths, Knowl. Inf. Syst. 45 (1) (2015) 105–132.

[6] A. Vahdatpour, N. Amini, M. Sarrafzadeh, Toward unsupervised activity discovery using multi-dimensional motif detection in time series., in: IJCAI, 9, 2009, pp. 1261–1266.

[7] J. Lin, E. Keogh, S. Lonardi, P. Patel, Finding motifs in time series(2002).

[8] E. Fuchs, T. Gruber, J. Nitschke, B. Sick, On-line motif detection in time series with swiftmotif, Pattern Recognit. 42 (11) (2009) 3015–3031.

[9] A.M. Azmi, A.M. Al-Ssulami, Discovering common recurrent patterns in multiple strings over large alphabets, Pattern Recognit. Lett. 54 (2015) 75–81.

[10] F. Zhou, F. De la Torre, J.F. Cohn, Unsupervised discovery of facial events, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2574–2581.

[11] D. Batra, A. Kowdle, D. Parikh, J. Luo, T. Chen, icoseg: Interactive co-segmentation with intelligent scribble guidance, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3169–3176.

[12] A. Joulin, K. Tang, L. Fei-Fei, Efficient image and video co-localization with frank-wolfe algorithm, in: European Conference on Computer Vision (ECCV), Springer, 2014, pp. 253–268.

[13] W.-S. Chu, Y. Song, A. Jaimes, Video co-summarization: video summarization by visual co-occurrence, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[14] W.-S. Chu, F. De la Torre, J.F. Cohn, D.S. Messinger, A branch-and-bound framework for unsupervised common event discovery, Int. J. Comput. Vision 123 (3) (2017) 372–391, doi:10.1007/s11263-017-0989-7.

[15] H. Stern, M. Shmueli, S. Berman, Most discriminating segment–longest common subsequence (mdslcs) algorithm for dynamic hand gesture classification, Pattern Recognit Lett. 34 (15) (2013) 1980–1989.

[16] R. Goldenberg, R. Kimmel, E. Rivlin, M. Rudzsky, Behavior classification by eigendecomposition of periodic motions, Pattern Recognit. 38 (7) (2005) 1033–1043.

[17] A. Briassouli, N. Ahuja, Extraction and analysis of multiple periodic motions in video sequences, IEEE Trans. Pattern Anal. Mach. Intell. 29 (7) (2007) 1244–1261.

[18] W. Ren, S. Singh, M. Singh, Y.S. Zhu, State-of-the-art on spatio-temporal information-based video retrieval, Pattern Recognit. 42 (2) (2009) 267–282.

[19] J. Song, L. Gao, L. Liu, X. Zhu, N. Sebe, Quantization-based hashing: a general framework for scalable image and video retrieval, Pattern Recognit. 75 (2018) 175–187.

[20] E. Ramasso, C. Panagiotakis, D. Pellerin, M. Rombaut, Human action recognition in videos based on the transferable belief model, Pattern Anal. Appl. 11 (1) (2008) 1–19.

[21] S. Sempena, N.U. Maulidevi, P.R. Aryan, Human action recognition using dynamic time warping, in: Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, IEEE, 2011, pp. 1–5.

[22] P. Foggia, B. Gauzere, A. Saggese, M. Vento, Human action recognition using an improved string edit distance, in: Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on, IEEE, 2015, pp. 1–6.
[23] L. Brun, G. Percannella, A. Saggese, M. Vento, Action recognition by using kernels on aclets sequences, Comput. Vision Image Understanding 144 (2016) 3–13.
[24] D. Gong, G. Medioni, X. Zhao, Structured time series analysis for human action segmentation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1414–1427.
[25] M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, A. Del Bimbo, Motion segment decomposition of rgb-d sequences for human behavior understanding, Pattern Recognit. 61 (2017) 222–233.
[26] G. Guo, A. Lai, A survey on still image based human action recognition, Pattern Recognit. 47 (10) (2014) 3343–3361.
[27] M. Ziaeefard, R. Bergevin, Semantic human activity recognition: a literature review, Pattern Recognit. 48 (8) (2015) 2329–2345.
[28] L. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Chapter 4, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
[29] M. Morel, C. Achard, R. Kulpa, S. Dubuisson, Time-series averaging using constrained dynamic time warping with tolerance, Pattern Recognit. 74 (2018) 77–89.
[30] F. Zhou, F. De la Torre, Generalized canonical time warping, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 279–294.
[31] A.S. Park, J.R. Glass, Unsupervised pattern discovery in speech, IEEE Trans. Audio Speech Lang. Process. 16 (1) (2008) 186–197.
[32] G. Trigeorgis, M. Nicolaou, S. Zafeiriou, B. Schuller, Deep canonical time warping for simultaneous alignment and representation learning of sequences, IEEE Trans. Pattern Anal. Mach. Intell. (2017), doi:10.1007/s10115-015-0898-4.
[33] J. Serra, J.L. Arcos, Particle swarm optimization for time series motif discovery, Knowl. Based Syst. 92 (2016) 127–137.
[34] Y. Shou, N. Mamoulis, D.W. Cheung, Fast and exact warping of time series using adaptive segmental approximations, Mach. Learn. 58 (2) (2005) 231–267, doi:10.1007/s10994-005-5828-3.
[35] R. Emonet, J. Varadarajan, J.-M. Odobez, Temporal analysis of motif mixtures using dirichlet processes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (1) (2014) 140–156.
[36] D. Minnen, T. Starner, I. Essa, C. Isbell, Improving activity discovery with automatic neighborhood estimation, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, in: IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 2814–2819.
[37] B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 493–498.
[38] A.S. Park, J.R. Glass, Unsupervised pattern discovery in speech, IEEE Trans. Audio Speech Lang. Process. 16 (1) (2008) 186–197.
[39] C. Rother, T. Minka, A. Blake, V. Kolmogorov, Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, IEEE, 2006, pp. 993–1000.
[40] A. Joulin, F. Bach, J. Ponce, Multi-class cosegmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 542–549.
[41] J.C. Rubio, J. Serrat, A. López, N. Paragios, Unsupervised co-segmentation through region matching, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 749–756.
[42] A. Faktor, M. Irani, Co-segmentation by composition, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1297–1304.
[43] J.C. Rubio, J. Serrat, A. López, Video co-segmentation, in: Asian Conference on Computer Vision, Springer, 2012, pp. 13–24.
[44] D.-J. Chen, H.-T. Chen, L.-W. Chang, Video object cosegmentation, in: 20th ACM International Conference on Multimedia, in: MM '12, ACM, New York, NY, USA, 2012, pp. 805–808, doi:10.1145/2393347.2396317.
[45] W.-C. Chiu, M. Fritz, Multi-class video co-segmentation with a generative multi-video model, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.
[46] L. Wang, G. Hua, R. Sukthankar, J. Xue, N. Zheng, Video object discovery and co-segmentation with extremely weak supervision, in: European Conference on Computer Vision (ECCV), 2014.
[47] X. Zhang, M.H. Mahoor, Task-dependent multi-task multiple kernel learning for facial action unit detection, Pattern Recognit. 51 (2016) 187–196.
[48] D. Yeo, B. Han, J.H. Han, Unsupervised co-activity detection from multiple videos using absorbing markov chain, in: 30th AAAI Conference on Artificial Intelligence, AAAI Press, 2016, pp. 3662–3668.
[49] J. Yuan, J. Meng, Y. Wu, J. Luo, Mining recurring events through forest growing, IEEE Trans. Circuits Syst. Video Technol. 18 (11) (2008) 1597–1607.
[50] O. Levy, L. Wolf, Live repetition counting, in: IEEE International Conference on Computer Vision, 2015, pp. 3020–3028.
[51] S. Shariat, V. Pavlovic, Robust time-series retrieval using probabilistic adaptive segmental alignment, Knowl. Inf. Syst. 49 (1) (2016) 91–119, doi:10.1007/s10115-015-0898-4.
[52] C.H. Lampert, M.B. Blaschko, T. Hofmann, Efficient subwindow search: a branch and bound framework for object localization, IEEE Trans. Pattern Anal. Mach. Intell. 31 (12) (2009) 2129–2142, doi:10.1109/TPAMI.2009.144.
[53] D. Johnson, Efficient algorithms for shortest paths in sparse networks, J. ACM 24 (1) (1977) 1–13.
[54] Y. Zhu, Z. Zimmerman, N.S. Senobari, C.-C.M. Yeh, G. Funning, A. Mueen, P. Brisk, E. Keogh, Matrix profile ii: exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins, in: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE, 2016, pp. 739–748.
[55] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley mhad: a comprehensive multimodal human action database, in: Applications of Computer Vision (WACV), 2013 IEEE Workshop on, IEEE, 2013, pp. 53–60.
[56] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Mach. Vision Appl 24 (5) (2013) 971–981, doi:10.1007/s00138-012-0450-4.
[57] D.P. Doane, L.E. Seward, Measuring skewness: a forgotten statistic, J. Stat. Educ. 19 (2) (2011) 1–18.

Costas Panagiotakis received the B.A, M.Sc., and Ph.D. degrees from the Dep. of Computer Science, University of Crete, Greece, in 2001, 2003, and 2007, respectively. He is Associate Professor with the Dep. of Business Administrator (Agios Nikolaos), TEI of Crete and a researcher at the Institute of Computer Science (ICS), Foundation for Research and Technology-Hellas (FORTH) in Heraklion, Crete, Greece. He is the author of one book and more than 60 articles in international journals and conferences. His interests include image analysis, pattern recognition, multimedia and signal processing. For more information, please visit www.csd.uoc.gr/~cpanag.

Konstantinos Papoutsakis received the B.A, M.Sc from the Dep. of Computer Science, University of Crete, Greece, in 2007 and 2010, respectively. Currently, he is Ph.D student at the Computer Science Department of University of Crete in Heraklion, Greece and a postgraduate scholar at the Computational Vision & Robotics Lab (CVRL) at the Institute of Computer Science of FORTH. His interests include computer vision, machine learning and robotics, especially topics related to human motion analysis, action understanding, human-robot interaction and vision for robots. For more information, please visit http://users.ics.forth.gr/~papoutsa.

Antonis Argyros is a Professor of Computer Science at the Computer Science Department (CSD), University of Crete (UoC) and a researcher at the Institute of Computer Science (ICS), Foundation for Research and Technology-Hellas (FORTH) in Heraklion, Crete, Greece. His research interests fall in the areas of computer vision with emphasis on tracking, human gesture and posture recognition, 3D reconstruction and omnidirectional vision. He is also interested in applications of computer vision in the fields of robotics and smart environments. In these topics, he has published more than 120 papers in peer reviewed scientific journals, conferences and workshops. Home page: http://users.ics.forth.gr/~argyros.