Temporal Action Co-segmentation in 3D Motion Capture Data and Videos

Konstantinos Papoutsakis[1,2] , Costas Panagiotakis[1,3] , Antonis A. Argyros[1,2]

[1] Computational Vision and Robotics Laboratory, Institute of Computer Science, FORTH, Greece
[2] Computer Science Department, University of Crete, Greece
[3] Business Administration Department (Agios Nikolaos), TEI of Crete, Greece
{papoutsa,cpanag,argyros}@ics.forth.gr

Abstract

Given two action sequences, we are interested in spotting/co-segmenting all pairs of sub-sequences that represent the same action. We propose a totally unsupervised solution to this problem. No a-priori model of the actions is assumed to be available. The number of common sub-sequences may be unknown. The sub-sequences can be located anywhere in the original sequences, may differ in duration and the corresponding actions may be performed by a different person, in different style. We treat this type of temporal action co-segmentation as a stochastic optimization problem that is solved by employing Particle Swarm Optimization (PSO). The objective function that is minimized by PSO capitalizes on Dynamic Time Warping (DTW) to compare two action sub-sequences. Due to the generic problem formulation and solution, the proposed method can be applied to motion capture (i.e., 3D skeletal) data or to conventional RGB videos acquired in the wild. We present extensive quantitative experiments on standard data sets as well as on data sets we introduced in this paper. The obtained results demonstrate that the proposed method achieves a remarkable increase in co-segmentation quality compared to all tested state of the art methods.

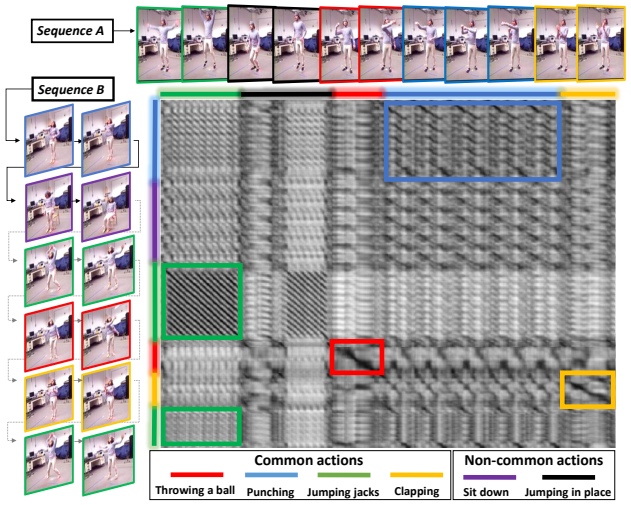


Figure 1: Given two image sequences that share common actions, our goal is to automatically co-segment them in a totally unsupervised manner. In this example, there are four common actions and two non-common actions. Notice that there are two instances of the 1st action of sequence A in sequence B. Each point of the grayscale background encodes the pairwise distance of the corresponding sequence frames.

1. Introduction

The unsupervised discovery of common patterns in images and videos is considered an important and unsolved problem in computer vision. We are interested in the temporal aspect of the problem and focus on action sequences (sequences of 3D motion capture data or video data) that contain multiple common actions. The problem was introduced in [10] as Temporal Commonality Discovery (*TCD*). Our motivation and interest to the problem stems from the fact that the discovery of common action patterns in two or more sequences provides an intuitive as well as efficient way to segment them, to identify a set of elementary actions performed in the sequences and, at a higher level, to build models of the performed actions in an unsupervised manner.

We propose a novel method for solving this problem that operates on multivariate time-series (i.e., a feature vector of fixed dimensionality per frame), representing action-relevant information. Assuming that two sequences/time-series contain a number of common action sub-sequences, we aim at identifying those pairs in a unsupervised manner, in the sense that no prior models, no information on the lengths and no labels of the actions are available. As illustrated graphically in Fig. 1, a shared (common) sub-sequence may appear anywhere in both sequences and may

differ in duration as well as in the style of execution. Such a commonality is defined by four values: the start positions of the co-segmented sub-sequences and their possibly different lengths, in both sequences. We cast the search for a single such commonality as a stochastic optimization problem that is solved based on Particle Swarm Optimization (PSO) [20]. Essentially, PSO searches for a commonality that minimizes the dissimilarity between the co-segmented sub-sequences. This is quantified by the cost of their non-linear alignment through Dynamic Time Warping (DTW) [35]. Iterative invocations of this optimization process identify all commonalities. Their number can either be known a-priori, or can be automatically identified, leading to a supervised or unsupervised variant of our method, respectively.

Experiments were carried out on several datasets that contain sequences of human or animal actions in lab settings or in-the-wild. The datasets involve either motion capture data or conventional RGB videos. The quantitative analysis of the obtained results reveals that the proposed strategy improves the co-segmentation overlap metric by a great margin over the best competing state of the art methods.

2. Related work

The term *co-segmentation* was originally introduced in computer vision by Rother et al. [38], as the task of joint segmentation of "something similar" given a set of images. This idea aspires to eliminate the need for tedious supervised training, enabling unsupervised or weakly supervised solutions to a number of interesting problems, such as automatic annotation of human actions in videos [11, 4].

Image/object co-segmentation: Several state-of-art methods have been proposed for co-segmenting similar image regions in images [27, 30, 5] and for object co-segmentation by extracting one or multiple prominent object(s) given an image pair [45, 18] or a single image [13]. The method proposed in [40] performs unsupervised co-segmentation of multiple common regions of objects in multiple images.

Video co-segmentation: Recently, the idea was extended to video segmentation towards common fore/background segmentation [39] and single or multiple object co-segmentation [6, 44]. In [8], multi-class video object co-segmentation is performed even when the number of object classes and object class instances are unknown. These works assume that all frames of videos contain the target object(s). This assumption is relaxed in [49], where the target objects are jointly discovered and co-segmented in multiple videos even if they do not appear in some frames.

Unsupervised segmentation of time-series: Several methods deal with the problem of finding one or multiple common temporal patterns in a single or multiple sequences [29, 7]. A solution to this problem is useful in various broad domains ranging from bioinformatics and economics to computer science and engineering. Various methods deal with the problem using temporal clustering [54], segmentation [21], temporal alignment [53], etc. In a recent work, Wang et al. [50] proposed a method for unsupervised temporal segmentation of human repetitive actions based on motion capture data. This is achieved by analyzing frequencies of kinematic parameters, detecting zero-velocity crossings and clustering of sequential data. Recently, the method in [2] studied the use of convolutional auto-encoders for unsupervised mining recurrent temporal patterns mixed in multivariate time series. The approach presented in [26] also uses CNNs to analyze sequentially blocks of 20 non-consecutive frames in an input video sequence in order to finally count the repetitions of approximately the same action in an online manner. The method in [12] is able to automatically determine the number of motifs shared by one or more video or audio sequences, find where they appear in each sequence and determine their lengths. Dynamic Time Warping (DTW) [35] is widely-used for temporal non-linear alignment of two sequences of different lengths for temporal alignment of human motion [53] or unsupervised speech processing [33]. In [33], a segmental variant of DTW was proposed to discover an inventory of lexical speech units in an unsupervised manner.

Action co-segmentation: The method in [15] performs common action extraction in a pair of videos by segmenting the frames of both videos that contain the common action. The method relies on measuring the co-saliency of dense trajectories on spatio-temporal features. The method proposed in [54] can discover facial units in video sequences of one or more persons in an unsupervised manner using temporal segmentation and clustering of facial features. Another recent work [9] tackles the problem of video co-summarization, by exploiting visual co-occurrence of the same topic across multiple videos, using a Maximal Biclique Finding (MBF) algorithm to discover one or multiple commonalities between and within two long action sequences. The work in [52] introduces an unsupervised learning algorithm using absorbing Markov chain in order to detect a common activity of variable length from a set of videos or multiple instances of it in a single video. Our work is most relevant to the Temporal Commonality Discovery (*TCD*) method [10] that discovers common semantic temporal patterns in a pair of videos or time-series, in an unsupervised manner. This is treated as an integer optimization problem that uses the branch-and-bound (B&B) algorithm [25] for searching for an optimal solution over all possible segments in each video sequence.

Our contribution: We present a novel solution to the problem of temporal action co-segmentation. The proposed method is totally unsupervised and assumes a very general representation of its input. Moreover, it is shown to outperform *TCD* and other state of the art methods by a large margin, while achieving a better computational performance.

3. Method description

Our approach consists of four components that address (a) feature extraction and representation of the data (Section 3.1), (b) DTW-based comparison of sequences of actions (Section 3.2) (c) an objective function that quantifies a potential sub-sequence commonality (Section 3.3) and (d) the use of (c) in an evolutionary optimization framework for spotting one common sub-sequence (Section 3.4) and all common sub-sequences (Section 3.5) in a pair of sequences.

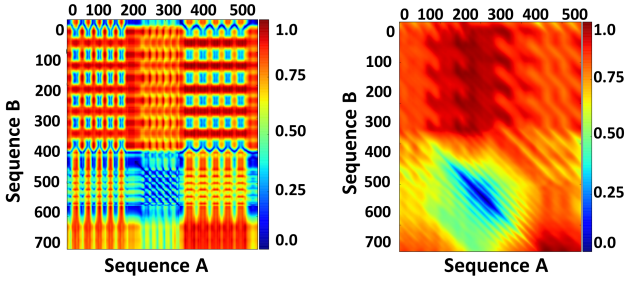
3.1. Features and representation

The proposed framework treats sequences of actions as multivariate time-series. More specifically, it is assumed that the input consists of two action sequences S_A and S_B of lengths l_A and l_B, respectively. Each "frame" of such a sequence is encoded as a feature vector $f \in \mathbb{R}^d$. The generality of this formulation enables the consideration of a very wide set of features. For example, vectors f can stand for motion-capture data representing the joints of a human skeleton or a Bag-of-Features representation of appearance and motion features based on dense trajectories [47, 43] extracted from conventional RGB videos. Section 4 presents experiments performed on four datasets, each of which considers a different representation of human motion capture or image sequence features. This demonstrates the generality and the wide applicability of the proposed solution.

3.2. Comparing action sequences based on DTW

A key component of our approach is a method that assesses quantitatively the similarity between two action sequences being treated as time-series, based on the notion of their temporal alignment. In a recent study, Wang et al. [51] performed an extended study on comparing 9 alignment methods across 38 data sets from various scientific domains. It was shown that the DTW algorithm [3, 35, 42], originally developed for the task of spoken word recognition [41], was consistently superior to the other studied methods. Thus, we adopt DTW as an alignment and comparison tool for pairs of sub-sequences within two sequences.

More specifically, given two action sequences S_A and S_B of lengths l_A and l_B, we first calculate the distance matrix $W_{A,B}$ of the pair-wise distances of all frames of the two sequences, as shown in Fig.2a. Depending on the nature of the sequences, different distance functions can be employed (e.g., Euclidean norm for motion capture data, χ^2 distance for histograms). The matrix $W_{1,2}$ represents the replacement costs, that is the input to the DTW method [3]. The algorithm calculates a cost matrix with a minimum distance/cost warp path traced through it, providing pairwise correspondences that establish a non-linear matching among all frames of the two sequences. The cumulative cost of all values across the path provides the alignment cost of



(a) Distance matrix (b) $F(p)$

Figure 2: (a) The pairwise distance matrix for all frames of sequences S_A, S_B. (b) Illustrating the scores of the objective function $F(p)$ (Eq. 1) for all possible starting points and sub-sequence lengths (see Sec.3.3 for details).

the two sequences, noted as D. The resulting minimum distance path contains a number n_p of diagonal elements that identify matches of frames between the two sequences. If the input sequences were identical time series of length l, the warp path through the cost matrix would be its diagonal, with $D = 0$ and $n_p = l$.

3.3. Evaluating a candidate commonality

Having a method for measuring the alignment cost of two sequences, our goal is now to define an effective objective function to evaluate candidate commonalities. A candidate commonality p is fully represented by the quadruple $p = (s_a, l_a, s_b, l_b)$, where s_a, s_b are the starting frames and l_a, l_b are the lengths of two sub-sequences of S_A and S_B, respectively. A commonality p can be viewed as a rectangle $R(p)$ whose top-left corner is located at point (s_a, s_b) and whose side lengths are l_a, l_b (e.g., see any colored rectangle in Fig. 1). We are interested in promoting commonalities p of low alignment cost (or equivalently, of high similarity) that also correspond to as many temporally matched frames as possible. To this end, we define an objective function $F(p)$ quantifying the quality of a possible commonality p:

$$F(p) = \frac{D(p) + c}{n_p(p) + 1}. \quad (1)$$

The quadruple p represents two sub-sequences of the original sequences, $D(p)$ is their DTW-based alignment cost and $n_p(p)$ is the number of matched frames of these subsequences. Essentially, $F(p)$ calculates the average alignment cost across the alignment path calculated using the DTW, by dividing the temporal alignment cost $D(p)$ with the number of matched frames $n_p(p)$. c is a small constant that guarantees that longer commonalities are favored over small ones, even in the unlikely case that $D(p) = 0$.

Figures 2a, 2b provide some intuition regarding the objective function and its 4D domain, based on a given pair of sequences. We assume that the two sequences S_A, S_B con-

tain a single common action across frames $[200..350]$ and $[400..600]$, respectively. Figure 2a illustrates the pairwise distance matrix $W_{A,B}$ of all their frames, calculated based on the Euclidean distance of the feature vectors representing the frames. Each cell (i, j) in the map of Fig. 2b represents the minimum objective function score F for two subsequences starting at frame i in S_A and j in S_B over all possible combinations of allowed lengths. Thus, the presented map visualizes the 4D parametric space by a 2D map of the responses of $F(\cdot)$. It can be verified that low scores are concentrated near the area of the common sub-sequences.

3.4. Spotting a single commonality

Spotting a single commonality amounts to optimizing Eq. (1) over all possible commonalities p. In notation, the optimal commonality p^* is defined as:

$$p^* = \arg\min_p F(p). \tag{2}$$

Searching exhaustively all candidate solutions in the 4D parameter space spanned by possible commonalities is prohibitively expensive. We choose to consider this as a stochastic optimization problem that is solved based on the canonical Particle Swarm Optimization (PSO) algorithm [20, 19, 16], a powerful and versatile evolutionary optimization method. PSO is a derivative-free optimization method that handles multi-modal, discontinuous objective functions with several local minima. Optimization is performed through the evolution of particles (candidate solutions) of a population (swarm). Particles lie in the parameter space of the objective function to be optimized and evolve through a limited number of generations (iterations) according to a policy which emulates "social interaction". The main parameters of PSO are the number of particles and generations, the product of which determines its computational budget (i.e., the number of objective function evaluations). PSO and other meta-heuristic methods such as Simulated Annealing [22] and Differential Evolution [34] are not guaranteed to convergence to a globally optimal solution. However, in practice, PSO and its variants are efficient and achieve near-optimal solutions. Thus, PSO has been applied with success in several challenging, multidimensional optimization problems in computer vision such as 3D pose estimation and tracking of hands [31, 32], hands and objects [24] and humans [17].

For single action co-segmentation, PSO operates on the 4D space of all possible commonalities. Certain constraints apply to the parameters s_a, s_b, l_a and l_b. Specifically, it holds that $l_a, l_b \geq l_{min}$ $l_a, l_b \leq l_{max}$, $s_a \leq l_A - l_{min}$, $s_b \leq l_B - l_{min}$, where l_{min}, l_{max} are user-defined minimum/maximum allowed commonality lengths. The 4D search space of PSO is constrained accordingly. For each run of PSO, the particles are initialized randomly (uniform distribution) in valid positions of the search space. The decision on the actual number of particles and generations needed to come up with an accurate solution to the problem is taken based on experimental evidence and is discussed in Section 4. Finally, sequences do not necessarily contain a commonality. Such situations can be identified by noting that the minimization of Eq. 2 results in a large value.

3.5. Spotting multiple commonalities

A joint solution to the problem for identifying N commonalities in a single run of PSO would require to explore a $4N$-dimensional space. This can become intractable for PSO when considering large values of N. Thus, we resort to an iterative optimization procedure that identifies a single commonality at a time.

It should be noted that the n-th commonality p_n should overlap as less as possible with the $n-1$ previously identified commonalities p_i, $1 \leq i \leq n-1$. There is a fairly intuitive way of identifying this overlap. Given two commonalities p_i and p_j, their normalized intersection $\Omega(p_i, p_j)$ with respect to p_i is defined as:

$$\Omega(p_i, p_j) = \frac{|R(p_i) \cap R(p_j)|}{|R(p_i)|}, \tag{3}$$

where $R(p)$ is the region of a commonality p (see Section 3.3) and $|\cdot|$ measures the area of a 2D region. Given this, in order to identify the i-th commonality p_i, we define a new objective function that considers the DTW-based score of p_i (as before), but also its normalized intersection with the already identified commonalities. The optimal i-th commonality is thus defined as:

$$p_i^* = \arg\min_{p_i} \left(F(p_i) + \lambda \sum_{j=1}^{i-1} \Omega(p_j, p_i) \right), \tag{4}$$

where $\lambda > 0$ tunes the contribution of the two terms in the objective function. Large λ values exclude commonalities that have even a slight overlap with the already identified ones. In our implementation, we set $\lambda = 1$. Note that the objective function of Eq.(4) penalizes commonalities whose regions overlap but does not penalize non-overlaping commonalities that share rows (or columns) of the distance matrix. Thus, an action in a sequence can be matched with several instances of the same action in the second sequence.

Supervised vs unsupervised action co-segmentation: The iterative co-segmentation method described so far can be applied for a known number N of iterations, giving rise to N retrieved commonalities. We denote this variant of the algorithm as *S-EVACO* that stands for *Supervised EVolutionary Action CO-segmentation*. *S-EVACO* is useful when the number of the common action subsequences in a pair of

videos is known a-priori. However, a totally unsupervised method that does not assume this knowledge is definitely preferable. We approach the problem of developing such an unsupervised method as a model selection task. To this end, we consider a user-defined parameter that is the maximum possible number of common actions, noted as K. We run the iterative PSO-based optimization process for K iterations, retrieving K commonalities p_i as well as their fitness scores $F(p_i)$. We sort the commonalities in ascending order of $F(p_i)$. Finally, we consider all possible $K-1$ break points between the consecutive, sorted commonalities. We accept the break point j^* that maximizes the absolute difference of the mean values of fitness scores to the left and to the right of it. By doing so, we guarantee that the introduction of p_{j^*+1} into the commonalities solution set decreases substantially the quality of the solution. In notation,

$$j^* = \underset{j\in\{1,\ldots,K-1\}}{\arg\max} \left| \frac{1}{j}\sum_{i=1}^{j} F(p_i) - \frac{1}{K-j}\sum_{i=j+1}^{K} F(p_i) \right|. \quad (5)$$

The commonalities p_1 to p_{j^*} constitute the sought solution. We denote this variant of our method as *U-EVACO*.

4. Experimental evaluation

We assess the performance of the proposed action co-segmentation method and compare it with state-of-the art methods using various ground truthed datasets based on either 3D motion capture data or conventional (RGB) videos.

In a first series of experiments we investigate the computational budget (number of particles and generations) that is required by PSO to solve the co-segmentation problem. Typically, more particles and/or generations help PSO to better explore the parametric search space. However, beyond a certain point, the accuracy gains are disproportionally low compared to the increase in the computational requirements. These experiments lead to the selection of the computational budget that constitutes the best compromise between computational requirements and accuracy.

Then, we compare the resulting performance of the method based on the selected and fixed PSO budget with that of the state-of-the-art *TCD* method [10] the method proposed by Guo et al. [15] and our own implementation of two variants of the Segmental DTW [33]. In Segmental DTW, a local alignment procedure produces multiple warp paths having limited temporal variation and low distortion. Each warping path is limited to a diagonal region of a given width. The minimum length of each path is also given as a parameter. Since the Segmental DTW in an unsupervised method, we name our implementation of it as *U-SDTW*. We also consider a supervised variant, namely *S-SDTW*, where the number of common sub-sequences is known and identified by selecting the paths with the lower length-constrained minimum average distortion fragment [33]. The parameters of all competing methods were fine-tuned to optimize their performance. We report the best of the obtained results.

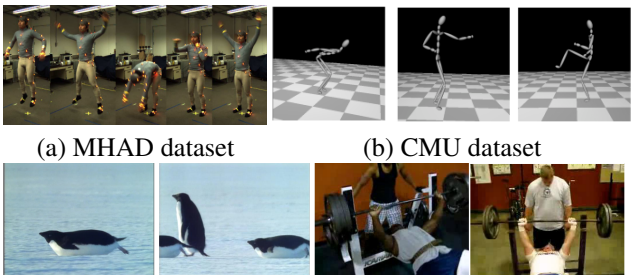
4.1. Datasets and performance metrics

The experimental evaluation was conducted using a total of 373 pairs of sequences, consisting of up to 2355 action sub-sequences and 1286 pairs of common actions. All the compiled datasets, the code for the proposed method and detailed optimized parameter settings for all methods used in the experiments are publicly available online[1].

MHAD101-s dataset: The Berkeley Multimodal Human Action Database (MHAD) [46] contains human motion capture data acquired by an optical motion capture system as well as conventional RGB video and depth data acquired from multiple views and depth sensors, respectively. All information streams are temporally synchronized and geometrically calibrated. The original dataset (see Fig. 3a) contains 11 actions performed by 12 subjects (7 male, 5 female). Each action is repeated 5 times by each subject. We considered all the available action categories except one (the action labeled as sit down/stand up), as it is a composition of the actions *No10-(sit down)* and *No11-(stand up)*. This alleviates potential problems and ambiguities in the definition of the ground truth. We selected only the first (out of the five) execution of an action by each subject, thus we collected a set of 120 action sequences. We use the motion capture (3D skeletal) data of the original dataset and we downsampled it temporally by a factor of 16 to reach the standard frame-rate of 30 fps. We then considered the subset of human actions defined above, as building blocks for synthesizing larger sequences of actions and defining pairs of such sequences for which ground truth regarding commonalities is available, by construction.

The resulting MHAD101-s dataset contains 101 pairs of action sequences. In 50 of the paired sequences, each sequence consists of 3 concatenated action clips and the paired sequences have exactly 1 in common. In 17 pairs, each sequence consists of 3-7 actions and the two sequences have 2 in common. In 17 pairs, each sequence consists of 3-7 actions and the paired sequences have 3 actions in common. Finally, in 17 pairs, each sequence consists of 4-7 actions and paired sequences have 4 in common. It is also guaranteed that (a) a sequence contains actions of the same subject (b) to promote style and duration variability, for every pair, the two sequences involve different subjects and (c) the placement of the common actions in the sequences is random. The lengths of a sequence and a common action range between 300 - 2150 and 55 - 910 frames, respectively.
Representing 3D motion capture data in MHAD101-s: Several representations of skeletal data have been proposed [23,

[1] http://www.ics.forth.gr/cvrl/evaco/



(a) MHAD dataset (b) CMU dataset

(c) 80-pair dataset

Figure 3: (a) Snapshots from the Berkeley MHAD dataset. (b) Snapshots from the CMU-Mocap dataset. (c) Pairs of snapshots from the 80-pair dataset.

28, 14, 37]. We employ a variant of the representation proposed in [37]. According to this, a 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, the representation is augmented with the 4 angles between the fore- and the back-arms as well as the angles between the upper- and lower legs.

CMU86-91 dataset: We also employed the CMU-Mocap database[2], originally presented in [1] (Fig. 3b) of continuous action sequences. We selected the set *Subject* 86 consisting of 14 labeled, long action sequences of skeletal-based human motion data, each consisting of up to 10 actions (between 4k-8k frames). In contrast to MHAD101-s, the actions are not concatenated but rather executed in a continuous manner. We utilize a variation of the original dataset, presented in Chu et.al [9] that concerns, (a) grouping of action labels to 24 categories of similar action (the original dataset consists of 48 pre-defined actions), (b) feature representation of human motion data based on the position and orientation of the skeletal root and relative joint angles that results in a 30-D feature vector per frame, (c) temporally down-sampled sequences by a factor of 4 to reach the standard frame-rate of 30 fps, (d) a set of 91 pairs of action sequences, by combining all individual sequences. The ground truth of each sequence is provided in [1]. We consider the median value of the three frame numbers provided as possible action boundaries for each action in a long sequence. We also note that the length of the sequences ranges between 330 and 1570 frames, and the length of their common action ranges between 70 and 1000 frames at 30 fps.

MHAD101-v dataset: The MHAD101-v dataset is identical to MHAD101-s in terms of action composition and co-segmentation-related ground truth. However, instead of employing the motion capture data stream, we employ the corresponding RGB video stream. Our motivation is to test the performance of the proposed method when it is fed with low-level video data. At the same time, the comparison of the performance of the method for the same sequences under completely different representations, provides interesting insights for both the representations and the method.

Representing video data in MHAD101-v: The employed representation is based on the Improved Dense Trajectories (IDT) features [47]. Based on the IDT, we compute four types of descriptors, namely trajectory shape, HOG, HOF, and MBH [47], using publicly available code[3], and the same configuration and parameters, as presented in [48, 47]. To encode features, we use a Bag-of-Features representation, separately for each type of descriptor and for each pair of videos in the dataset. More specifically, we built a codebook for each type of descriptors using k-means over the features extracted over the frames of the two videos of a pair. Then, we calculate the Bag-of-Features representation for each frame, which results in a per frame feature vector (histogram of frequencies of codewords) that captures information regarding the descriptors of the trajectories that were detected and tracked in a temporal window of 15 frames preceding that frame. We found that a codebook of 25 codewords is sufficient for our purposes. Finally, we concatenate all feature vectors calculated for each type of descriptors per frame in a single 100-D feature vector.

80-pair dataset: We also employ the publicly available [4] 80-pair dataset, specifically designed for the problem of common action extraction in videos and presented in the work of Guo et.al. [15]. Among the 80 pairs of the dataset, 50 are segmented clips of human actions from the UCF50 dataset [36] and 30 pairs are selected from BBC animal documentaries depicting animal actions in the wild. Thus, the dataset contains videos of continuous actions executed in unconstrained settings and environments.

Representing video data in the 80-pair: Dense point trajectories [43] based on optical flow are employed and encoded using MBH descriptors [47], following the same experimental setup as in [15] based on the publicly available code [5]. Then, the per-frame features are computed based on the publicly available code [4] of the method [15]. Indicatively, a motion based figure-ground segmentation is applied to each video to remove background trajectories and a Bag-of-Features representation based on the MBH descriptors of all frames for a pair of videos is employed using 25 codewords. Thus, 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.

[2] http://mocap.cs.cmu.edu/

[3] http://lear.inrialpes.fr/people/wang/

[4] www.lizhuwen.com/

[5] http://lmb.informatik.uni-freiburg.de

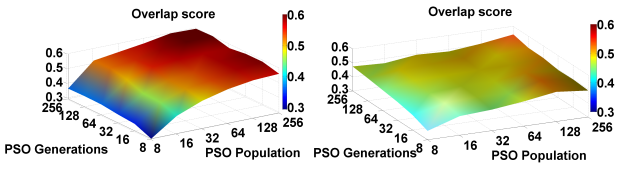


Figure 4: The overlap score of the objective function as a function of PSO particles and generations for the MHAD101-s (left) and MHAD101-v (right) datasets.

Performance metrics: In order to assess the performance of the evaluated methods we employed the standard metrics of precision $\mathcal{P}$, recall $\mathcal{R}$, $F1$ score and overlap $\mathcal{O}$ (intersection-over-union) [10]. In our context, 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. For each dataset, we calculate the average for each metric for all pairs.

4.2. Choosing the PSO budget

The effectiveness of the PSO optimization as well as its running time is determined by its number of particles s (candidate solutions) and generations g (particle evolutions). The product $s \cdot g$ equals the number of objective function evaluations. g and s are experimentally defined, so that the trade off between quality of solutions and running time is appropriately set. In that direction, we applied our method to the MHAD101-s and MHAD101-v datasets, running over all combinations of s, g in $\{8, 16, 32, 64, 128, 256\}$ i.e., from a lowest of $8 \times 8 = 64$ to a highest of $256 \times 256 = 65536$ objective function evaluations. For each combination, we considered the average overlap score of 5 runs.

Figure 4 summarizes the obtained results for the two datasets. In both, the overlap score increases as the PSO budget increases. Additionally, in both, the overlap score increases faster when increasing the number of particles than increasing the number of generations. Finally, while best performances do not differ considerably, optimization based on skeletal data appears easier that optimization based on video data. An overview of the 3D maps, provides evidence that combinations of at least 32-32 generations and particles achieve good results. The (g, s) configurations of (32, 128), (64, 128), (128, 32), (128, 64) achieve approximately the 90%, 95%, 90%, 95% of the maximum score achieved by the (256, 256) configuration, by using only the 6.25%, 12.5%, 6.25%, 12.5% of the maximum budget, respectively. We finally set (g, s) = (64, 128) in all our experiments. The reason is that generations need to be executed serially, while particles can be evaluated in parallel. Thus, the (64, 128) configuration can eventually be two times faster than the (128, 64) one.

Table 1: Evaluation results on the MHAD-101s and the CMU86-91 datasets involving 3D skeletal-based data.

MHAD101-s	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F1(\%)$	$\mathcal{O}(\%)$
TCD [10]	16.7	18.1	13.8	8.5
S-SDTW [33]	61.6	47.1	48.5	35.9
U-SDTW [33]	65.8	45.5	47.7	35.1
S-EVACO	**77.9**	**67.6**	**71.3**	**59.4**
U-EVACO	71.3	63.9	63.3	50.3
CMU86-91	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F1(\%)$	$\mathcal{O}(\%)$
TCD [10]	30.9	51.3	38.0	24.1
S-SDTW [33]	44.9	20.9	27.6	16.1
U-SDTW [33]	44.9	20.9	27.6	16.1
S-EVACO	67.6	**77.1**	**71.6**	**57.5**
U-EVACO	**71.3**	67.4	65.2	51.0

4.3. Action co-segmentation on skeletal data

Results on MHAD101-s: We allow all methods to search for common sub-sequences whose lengths varies in the range $[25..1370]$ (from half the length of the shortest action to 1.5 times the length of the largest one). Results are reported in Table 1. The scores of the methods are presented as $\%$ average scores in the tables, computed over all individual scores per sample (pairs of sequences) of a dataset. The scores of the proposed *S-EVACO* and *U-EVACO* methods are the average scores over all samples of a dataset computed after 10 repetitions of the experiment for each dataset. *S-EVACO* achieves an overlap score of 59.4% and outperforms *TCD* by over 50% and both variants of Segmental DTW (*U-SDTW*/*S-SDTW*) by over 20% for all the reported metrics. The overlap metric of *U-EVACO* is 9% lower compared to that of *S-EVACO*. Still, we stress that the *unsupervised* version of the proposed method outperforms the state of the art *supervised* methods by a very wide margin.

Results on CMU86-91: We allow all methods to search for common sub-sequences whose lengths varies in the range $[70..1135]$ (from half the length of the shortest action, to 1.5 times the length of the largest one). Results for the CMU86-91 dataset are reported in Table 1. The proposed approaches outperform *TCD* [10] in all reported metrics ($27 - 33\%$ higher overlap). We also note the considerably higher performance of our method compared to *S-SDTW* and *U-SDTW* ($36 - 41\%$ higher overlap).

4.4. Action co-segmentation on video data

Results on MHAD101-v: The results are summarized in Table 2. Given the number of common actions to be discovered in each pair of sequences, the proposed *S-EVACO* method outperforms *S-SDTW* by over 20% and 10% with respect to the overlap and the other metrics, respectively.

Table 2: Evaluation results on the MHAD-101v and the 80-pair datasets involving video data.

MHAD101-v	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F1(\%)$	$\mathcal{O}(\%)$
TCD [10]	20.6	14.0	15.4	19.3
S-SDTW [33]	65.2	49.1	50.5	37.7
U-SDTW [33]	69.4	45.7	48.0	35.5
S-EVACO	**76.6**	**66.8**	**69.8**	**56.2**
U-EVACO	63.3	63.3	58.8	45.9
80-pair	$\mathcal{R}(\%)$	$\mathcal{P}(\%)$	$F1(\%)$	$\mathcal{O}(\%)$
TCD [10]	22.9	65.4	31.2	21.5
S-SDTW [33]	27.8	52.2	31.4	21.6
U-SDTW [33]	34.6	60.6	37.3	25.6
S-EVACO	**75.8**	77.2	**73.9**	**64.5**
U-EVACO	61.0	69.7	62.0	54.2
Guo [15]	55.6	**78.1**	60.9	51.6

TCD has a less than 20% overlap score, and its performance is lower by more than 30% with respect to recall, precision and F1 metrics. The unsupervised variant (*U-EVACO*) results in 13% less overlap compared to *S-EVACO*. In addition, *S-SDTW* achieves lower overlap, precision and F1 scores by 8%, 14% and 10%, compared to *S-EVACO*.

Results on 80-pair dataset: Table 2 summarizes the findings on this dataset. Besides *TCD*, *S-SDTW* and *U-SDTW*, we compare our approach to the method of Guo et al. [15]. We employed the publicly available implementation of that method and we run it with the parameters suggested in [15] for that dataset. The *TCD*, *S-SDTW* and *U-SDTW* methods achieve comparable scores, with *TCD* performing the lower scores in all metrics except the precision that is 5% better than that of *U-SDTW*. The proposed *S-EVACO*, *U-EVACO* have similar scores, mainly due to the fact that all pairs of videos in that dataset contain a single common action to be discovered. Both variants of the proposed method outperform *TCD*, *S-SDTW* and *U-SDTW* methods with over 40% improvements for the overlap and between 17% and 53% for the other metrics. Both proposed variants also score higher than Guo's method [15] in overlap (12% improvement), F1 score (12%) and recall (20%).

Figure 5 summarizes the findings in all datasets. The left column shows the $\%$ of pairs where the overlap is above a certain threshold. The right column shows plots of the mean $F1$ score for all sequence pairs, after zeroing the $F1$ score of pairs below an overlap threshold. Plots are shown for all pairs of the datasets (top), for those involving video data only (middle) and skeletal data only (bottom). It can be verified that the *S-EVACO* and *U-EVACO* outperform the state of the art by a large margin. In general, the BoW-based representation employed by TCD is orderless, so the comparison of actions misses important temporal content.

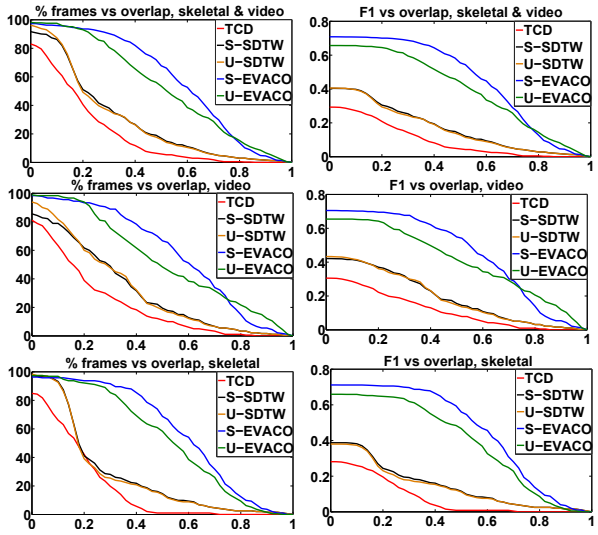


Figure 5: Summary of the obtained results in all datasets.

On the contrary, our DTW-based approach captures this important temporal dimension. Regarding execution time, *U-EVACO* requires, on average, 10s for processing a pair of videos of ~ 1000 frames each, discovering $8-10$ common action sub-sequences. This makes it slower but comparable to *U-SDTW* and more than two times faster than *TCD*.

5. Summary and conclusions

We presented a novel method for the temporal co-segmentation of all common actions in a pair of action sequences. We treated this as a stochastic optimization problem whose solution is the start positions and the lengths of the sub-sequences of the input sequences that define action segments of maximum similarity. Optimization was performed based on iterative Particle Swarm Optimization with an objective function defined based on the non-linear DTW alignment cost of two sub-sequences. The proposed approach operates on multivariate time series. As such, it can assume a variety of image/video/motion representations. Two variants were presented, one that assumes that the number of commonalities is known (*S-EVACO*) and one that does not require that information (*U-EVACO*). Both variants were extensively tested on challenging datasets of motion capture and video data, with a variety of features and representations and in comparison with state of the art methods. The results demonstrated that the proposed approach outperforms all state of the art methods in all data sets by a large margin.

Acknowledgements

This work was partially supported by the H2020 projects *ACANTO* and *Co4Robots*. The contributions of Damien Michel and Paschalis Panteleris, members of CVRL/ICS/FORTH are gratefully acknowledged.

References

[1] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004*, pages 185–194. Canadian Human-Computer Communications Society, 2004.

[2] K. Bascol, R. Emonet, E. Fromont, and J.-M. Odobez. *Unsupervised Interpretable Pattern Discovery in Time Series Using Autoencoders*, pages 427–438. Springer International Publishing, Cham, 2016.

[3] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[4] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, pages 628–643. Springer, 2014.

[5] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *IEEE ICCV*, pages 2579–2586, Nov 2011.

[6] D.-J. Chen, H.-T. Chen, and L.-W. Chang. Video object cosegmentation. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 805–808, New York, NY, USA, 2012. ACM.

[7] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 493–498, New York, NY, USA, 2003. ACM.

[8] W.-C. Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[9] W.-S. Chu, Y. Song, and A. Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *IEEE CVPR*, pages 3584–3592, 2015.

[10] W.-S. Chu, F. Zhou, and F. De la Torre. Unsupervised temporal commonality discovery. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *ECCV*, volume 7575 of *Lecture Notes in Computer Science*, pages 373–387. Springer Berlin Heidelberg, 2012.

[11] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *IEEE ICCV*, pages 1491–1498, Sept 2009.

[12] R. Emonet, J. Varadarajan, and J. M. Odobez. Temporal analysis of motif mixtures using dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):140–156, Jan 2014.

[13] A. Faktor and M. Irani. Co-segmentation by composition. In *IEEE ICCV*, pages 1297–1304, 2013.

[14] D. Gavrila. The visual analysis of human movement. *Comput. Vis. Image Underst.*, 73(1):82–98, Jan. 1999.

[15] J. Guo, Z. Li, L.-F. Cheong, and S. Z. Zhou. Video co-segmentation for meaningful action extraction. In *IEEE ICCV*, pages 2232–2239. IEEE, 2013.

[16] S. Helwig and R. Wanka. Particle swarm optimization in high-dimensional bounded search spaces. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 198–205, April 2007.

[17] V. John, S. Ivekovic, and E. Trucco. Articulated human motion tracking with hpso. In *VISAPP (1)*, pages 531–538, 2009.

[18] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *IEEE CVPR*, pages 542–549. IEEE, 2012.

[19] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[20] J. Kennedy, J. F. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.

[21] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296, 2001.

[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[23] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 559–568, New York, NY, USA, 2004. ACM.

[24] N. Kyriazis and A. Argyros. Scalable 3d tracking of multiple interacting objects. In *IEEE CVPR*, pages 3430–3437. IEEE, 2014.

[25] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2129–2142, Dec. 2009.

[26] O. Levy and L. Wolf. Live repetition counting. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3020–3028, Dec 2015.

[27] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1609–1616. IEEE, 2010.

[28] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, Nov. 2006.

[29] A. Mueen. Enumeration of time series motifs of all lengths. In *2013 IEEE 13th International Conference on Data Mining*, pages 547–556, Dec 2013.

[30] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *IEEE CVPR*, CVPR '11, pages 1881–1888, Washington, DC, USA, 2011. IEEE Computer Society.

[31] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, volume 1, page 3, 2011.

[32] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *IEEE CVPR*, pages 1862–1869. IEEE, 2012.

[33] A. S. Park and J. R. Glass. Unsupervised pattern discovery in speech. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1):186–197, 2008.

[34] K. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[35] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*, volume Chapter 4. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[36] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[37] I. Rius, J. Gonzlez, J. Varona, and F. X. Roca. Action-specific motion prior for efficient bayesian 3d human body tracking. *Pattern Recognition*, 42(11):2907 – 2921, 2009.

[38] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *IEEE CVPR*, volume 1, pages 993–1000. IEEE, 2006.

[39] J. C. Rubio, J. Serrat, and A. López. Video co-segmentation. In *ACCV*, pages 13–24. Springer, 2012.

[40] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *IEEE CVPR*, pages 749–756. IEEE, 2012.

[41] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

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

[43] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, pages 438–451, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[44] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *European Conference on Computer Vision*, pages 760–775. Springer, 2016.

[45] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *IEEE CVPR*, pages 2217–2224, June 2011.

[46] R. Vidal, R. Bajcsy, F. Ofli, R. Chaudhry, and G. Kurillo. Berkeley mhad: A comprehensive multimodal human action database. In *WACV*, WACV '13, pages 53–60, Washington, DC, USA, 2013. IEEE Computer Society.

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

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

[49] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, 2014.

[50] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy. Unsupervised temporal segmentation of repetitive human actions based on kinematic modeling and frequency analysis. In *3D Vision (3DV)*, pages 562–570. IEEE, 2015.

[51] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[52] D. Yeo, B. Han, and J. H. Han. Unsupervised co-activity detection from multiple videos using absorbing markov chain. In *30th AAAI Conference on Artificial Intelligence*, pages 3662–3668. AAAI Press, 2016.

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

[54] F. Zhou, F. D. la Torre, and J. F. Cohn. Unsupervised discovery of facial events. In *IEEE CVPR*, pages 2574–2581, June 2010.