Faculty of Informatics
Masaryk University

Content-Based Processing of Human Motion Data

Habilitation Thesis
(Collection of Works)

Jan Sedmidubský

March 2021

Abstract

Specialized hardware technologies or recent pose estimation software tools can digitize human motion into a discrete sequence of 3D skeletons. Such spatio-temporal data have enormous application potential in many fields, ranging from entertainment and sports to security and healthcare. To make the recorded data useful for a variety of applications, effective and efficient data management techniques are needed. This habilitation thesis introduces general-purpose techniques developed for classification, annotation, and searching in complex human skeleton data. The presented techniques are primarily based on recent advances in deep learning and similarity searching, with an emphasis on both effectiveness and performance issues. The applicability of selected techniques is also supported by developed prototype implementations or interactive web applications.

Acknowledgments

I want to thank all collaborators who have participated in the research papers of which I am the co-author. I would also like to thank the members of the Laboratory of Data Intensive Systems and Applications (DISA), led by prof. Pavel Zezula, for their long and fruitful research discussions. A special thank belongs to my family which had to experience all my successes and failures.

Contents

Part I

Commentary

Chapter 1

Introduction

Human motion can be digitized by estimating 3D positions of selected body points, typically *joints*, in time. Recorded joints captured at a given time moment form a *pose*, which can be visualized by a stick figure resembling a 3D *skeleton*. Therefore, human motion data are often denoted as 3D *skeleton sequences*. Traditionally, these spatio-temporal data have been captured using specialized hardware technologies, such as precise but expensive systems of synchronized optical cameras like Vicon, or cheap but inaccurate depth-sensor devices like Microsoft Kinect. A high-level overview of existing acquisition technologies is provided in Table 1.1. Today, there is a growing interest in developing pose-estimation software tools that are able to estimate joint positions from ordinary video data [1].

The acquired skeleton data have enormous application potential in a lot of domains [3, 4]. In *entertainment*, the data are used to render realistic-looking movements in movies, games, and virtual or augmented reality. This involves direct mapping of captured movements from live subjects to virtual characters and synthesizing animations in high quality [5, 6], or assisting people in learning dancing [7] or any movements according to a projected performance [8]. In *healthcare*, doctors and therapists can browse the recorded skeleton data to better determine the diagnosis and treatment of patients. Gait analysis helps determine neurodegenerative diseases [9], prevent possible injuries in the near future [10], evaluate different treatment outcomes for cerebral palsy [11], or identify individualized therapeutic strategies for running injuries [12]. A lot of research is devoted to rehabilitation systems that assist patients during recovery [13] and increase their engagement via gamification [14]. In professional *sports*, a research primarily focuses on posterior analysis and evaluation of athletic performances, e.g., in golf [15], dancing [16], figure-skating [17], or martial arts [18]. The skeleton data are also analyzed to predict the future tennis-shot direction [19], detect swimming strokes [20], or analyze the phases of long and triple jumps [21]. In *smart cities*, skeleton data from real-time sen-

Table 1.1: Acquisition technologies of 3D skeleton data.

Technology	Sensors	Frame rate	Occl. resist.[1]	Error margin	Cost	Mobility	Markers
Vicon[2] (optical sensors)	10–40	360	◐	mm	$$$	–	✓
xSens[3] (inertial sensors)	17	240	●	mm–cm	$$	✓	✓
Kinect v2[4] (RGB + depth)	2	30	○	cm	$	–	–
synchr. video cameras [2]	3	˜video	◐	cm	$	–	–
video + xNect [1]	1	˜video	○	>cm	$	✓	–

[1] Degree of resistance towards occlusions – resist. (●), partially resist. (◐), not resist. (○)
[2] https://www.vicon.com/ [3] https://www.xsens.com/
[4] https://developer.microsoft.com/en-us/windows/kinect/

sors and ordinary cameras can be used to analyze situations in crowded spaces, smart homes, or autonomous driving vehicles. This involves identification of subjects by posture and gait [22], customer analysis and shopping support [23], social-interaction understanding in public places [24], detection of abnormal activities of elderly people in smart homes [25], or movement prediction of pedestrians and cyclists approaching a camera in autonomous driving vehicles [26].

A great application potential together with the current progress in pose-estimation tools indicate a fast increase of 3D human motion data in the near future. To make the recorded data useful for a wide range of applications, we need general-purpose data management techniques that are able to effectively and efficiently analyze the motion content. Let us assume we have a set of skeleton sequences from a figure-skating competition. Then, a user can be interested in the following tasks: (i) categorizing the figure element performed in a given, manually selected motion segment, (ii) determining all occurrences of the triple Axel jump, or (iii) locating all competitors who performed a similar element as a specified query figure. These tasks are typically referred to as *action recognition*, *action detection*, and *search* or *subsequence search*.

Current research primarily focuses on processing of *actions*, which are short skeleton sequences with a clear semantics that is subjective to an observer (e.g., kick, punch, cartwheel, or Axel jump). The action recognition task aims at determining the class of pre-segmented actions based on a *labeled* set of training ones. This is typically solved using deep neural-network classifiers [27, 28, 29]. However, these classifiers can not be directly applicable to scenarios where skeleton data are captured as continuous *long motions* without any information about semantic partitioning. In such cases, the action detection task can be performed to determine the beginnings and endings of all occurrences of user-interested actions. This is usually solved by adapting recurrent neural networks [30, 31, 32]. The actions can even be predicted if early action detection is needed during

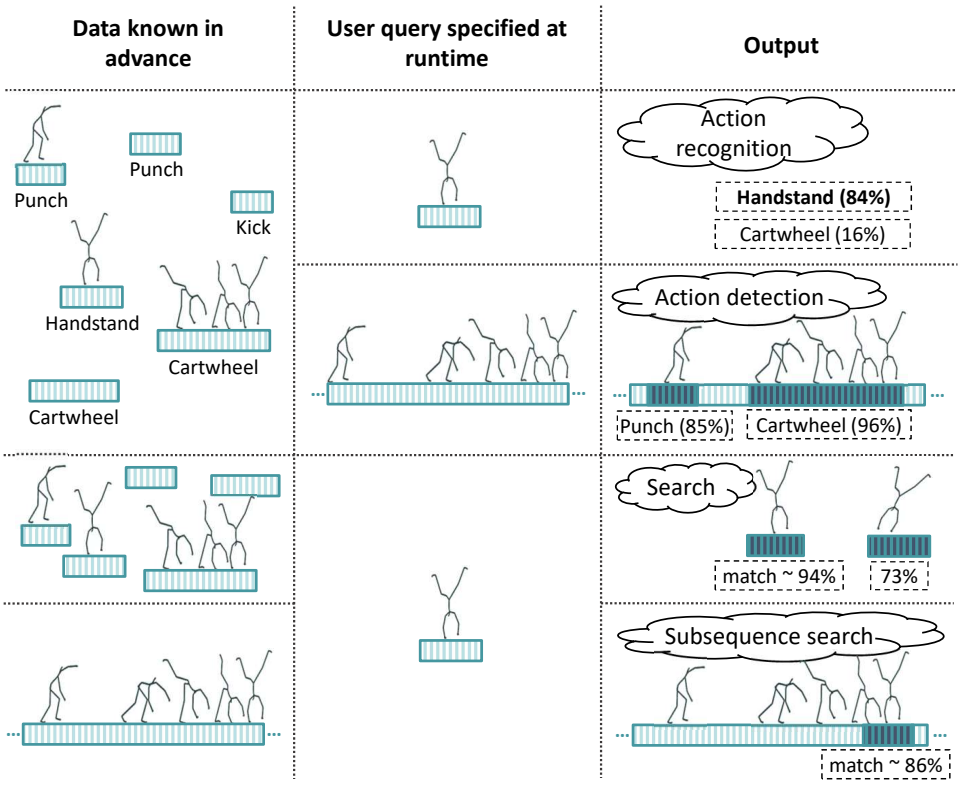


Figure 1.1: Basic motion-processing tasks: action recognition, action detection, search and subsequence search. The data known in advance represent a database that can be pre-processed offline, while a user query needs to be answered online.

online processing. The tasks of action recognition, detection or prediction require a set of labeled pre-segmented training actions to be specified in advance. If data labeling is not known, query-by-example search can be applied to inspect a collection of pre-segmented motions and find those that are the most similar to a specified query. If unsegmented and unlabeled long motions are only available, subsequence search can be used to retrieve the most similar sub-motions with respect to the query. All these tasks are graphically illustrated in Figure 1.1.

Challenges

The tasks of action recognition, action detection, search, and subsequence search are considered as the most useful for a wide range of applications. However, solving these tasks is difficult since they require completely different data-processing paradigms when compared to the traditional do-

mains such as attribute-like data, text or images. In addition, these tasks have to cope with variability, complexity, impreciseness, and voluminousness of the captured spatio-temporal data. This leads to the following two high-level challenges that are common for most of the motion-processing tasks.

- *Similarity modeling* – learning a metric for determining similarity between a pair of semantic actions, or any pieces of motions in general. The metric should preserve the motion semantics with respect to the needs of a target application, while being efficiently evaluated. The metric can be learned either in a supervised, or unsupervised way based on the availability of labeled training data.

- *Efficient processing* – organizing known data to be efficiently accessible during evaluation of user queries. This typically requires building various index structures with reasonable space requirements and applying approximate retrieval algorithms.

In the following, we describe how we have contributed to fulfilling the stated challenges from the perspective of the aforementioned tasks.

Scope of the Thesis

Since 2012, we have published 21 conference papers and 6 journal publications purely in the context of skeleton data processing. This thesis brings a brief commentary on these papers and highlights the 10 most significant works whose full versions can be found in the second part of this document (Part II: Work 1–Work 10). The mentioned papers are usually based on deep-learning and similarity-search principles and can be classified into the following topics:

1. *Metric learning* – determining a similarity between two skeleton sequences as a fundamental pre-requisite for any motion-processing task; the similarity is learned using deep neural networks [33, 34], or unsupervised feature-extraction approaches [35, 36, 37, 38];

2. *Gait recognition* – segmenting skeleton sequences semantically into gait cycles [39] whose specifically-defined similarity is used for identifying subjects based on the way they walk [40];

3. *Action recognition* – determining the classes of pre-segmented skeleton sequences using deep learning principles in combination with k-nearest neighbor classification [41, 42] and various data normalization and augmentation techniques [43, 44];

4. *Subsequence search* – searching for query-similar subsequences within long database motions; the database sequences are partitioned into short segments [45] whose features are efficiently organized [46, 47, 48, 49, 50, 51];

5. *Action detection* – annotating continuous skeleton sequences in both offline- and stream-based processing modes [52, 31].

We have also contributed to the field by developing additional cross-topic techniques that deal with general-purpose analysis of skeleton data. In particular, we have focused on analyzing quality of captured data [53], building a large dataset of continuous skeleton sequences [54], or estimating the accuracy gap between 2D and 3D skeleton modalities [55, 56].

The selected topics have also been summarized [57, 58] and presented as 3-hour tutorials at the top multimedia conferences – the ACM Multimedia (MM) 2018 and the ACM International Conference on Multimedia Retrieval (ICMR) 2019. The achieved results have additionally been recognized by other research community – the ESMAC (European Society for Movement analysis in Adults and Children) board has invited us to give a seminar lecture within the ESMAC 2018 conference. This conference belongs to one of the two largest world conferences on movement analysis in adults and children.

Except for the standard research papers, the proposed contributions have been additionally supported by developed prototype implementations or online web applications, some of them registered in the form of "software". Jan Sedmidubský is always the main author and developer of all the applications as well as software-based outputs.

Chapter 2

Proposed Content-Based Processing Techniques

This chapter briefly describes the proposed approaches structured according to the five topics mentioned above. The contributions achieved within these topics are confronted with state-of-the-art approaches and experimentally evaluated on different real-life datasets, e.g., HDM05 [59], CMU [1], PENN [60] or PKU-MMD [61]. Nevertheless, most of the experiments are conducted on the HDM05 dataset because it:

- Contains the highest number of 130 classes to be recognized, compared to other datasets having typically fewer than a half of classes;
- Provides only about 20 action samples for each class on average (minimum/maximum number of samples is $10/52$), compared to other datasets having one or two orders of magnitude more samples in each class;
- Provides not only segmented actions but also annotated long skeleton sequences that can be used for evaluation of subsequence-search or annotation algorithms.

Both the high number of classes and a limited number of samples in each class make the processing on the HDM05 dataset difficult, especially in the context of action recognition and action detection tasks.

2.1 Metric Learning

All the considered tasks – gait recognition, action recognition, action detection and subsequence search – explicitly or implicitly require to compare skeleton data based on *similarity*. It is important to realize that the exact

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

match on 3D skeleton sequences has very little meaning, as any motion can be hardly performed again in the exactly same way. The similarity is usually determined by pre-processing motion data to extract their application-specific *features* and comparing the extracted features by a *distance function*. Nevertheless, the extraction of high-quality features is very difficult since the similarity is subjective and context-dependent [35].

The former approaches have introduced many variants of *handcrafted* features, such as distances between joints [62], joint-angle rotations [63], or relational characteristics [64]. These features are commonly extracted for each motion pose in the form of n-dimensional vector (usually $n < 100$) and compared on the level of whole motions using expensive time-warping techniques, such as the Dynamic Time Warping (DTW) [64]. Except for time-consuming comparison, the handcrafted features have to be designed by domain experts and have limited ability to represent more complex dependencies in movement patterns. Therefore, the handcrafted features have been practically abandoned and replaced by *deep features* extracted from well-trained neural-network models [65].

Deep neural networks are often used for classification of actions into a predefined set of classes, typically using convolutional neural networks (CNN) [66, 67], graph convolutional networks (GCN) [68, 69], or Long Short-Term Memory (LSTM) networks [70, 71]. The learned parameters of hidden network layers can then be utilized for extraction of content-preserving features from input actions. Such features are often represented as fixed-size high-dimensional vectors (e.g., 4,096D features in [34]) and generalize very well when varied training data are provided. Contrary to the handcrafted features, the deep features have higher descriptive power and their fixed-size nature enables efficient and indexable comparison by the Manhattan or Euclidean distance functions.

In the following, we present our techniques for extraction of effective deep features using CNN and LSTM neural networks, and also the motion-word technique that transforms skeleton data into a compact text-like representation suitable for efficient indexing.

2.1.1 CNN Features

In [34], we have proposed a new approach for extraction of highly descriptive motion features using a fine-tuned deep convolutional neural network. First, we have encoded each 3D skeleton sequence into a 2D *motion image*. The colors of pixels within the motion image determine how the coordinates of individual joints change over time as the subject moves. Then, we have proposed to fine-tune the AlexNet convolutional neural network by the motion images of training actions. As the network is fine-tuned, the descriptive 4,096D feature vectors are extracted from the last hidden network layer, as schematically illustrated in Figure 2.1. The similarity of a pair of

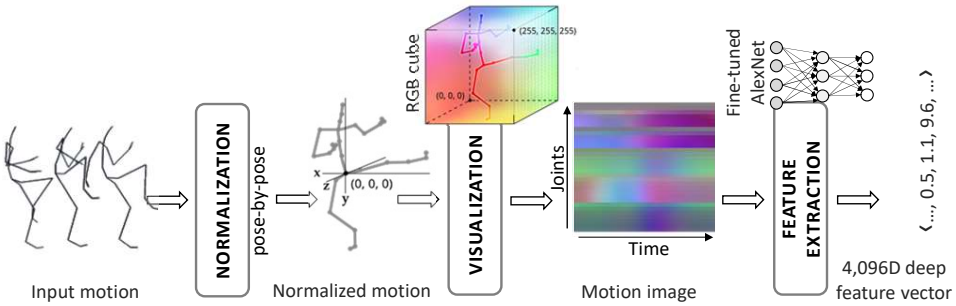


Figure 2.1: Illustration of the CNN feature extraction process outputting the fixed-size 4,096D vector for the input motion of a variable length.

motions is finally quantified by the Euclidean distance calculated between their corresponding deep feature vectors. The advantage of this approach is its tolerance towards an imprecise segmentation of training actions, the variance in movement speed, and a lower data quality in terms of precision of estimated 3D joint positions. More details about this approach can be found in the attached publication in Part II (Work 1). We have further proposed some improvements in generation of motion images [33], leading to slightly better descriptive power of the extracted deep features. In general, the motion image representation is convenient when a limited amount of training data is available. In such cases, small data amounts are sufficient when the pre-trained AlexNet is fine-tuned.

2.1.2 Bi-LSTM Features

The disadvantage of the CNN-based approach is that it assumes input data in the form of motion images that have to be resized into the fixed size before entering the AlexNet network, which leads to deformation of the temporal motion dimension. Therefore, in [43] we have proposed to adopt the LSTM variant of recurrent neural networks that well suit the sequential nature of motion data. Individual skeletons, represented as vectors of 3D joint coordinates, are gradually fed to recurrent network cells, and the hidden-state output of a previous time step is passed to the input of the current step. In particular, we have adopted a *bidirectional* LSTM (Bi-LSTM) neural network, which connects two hidden layers of opposite directions to the same output. As illustrated in Figure 2.2, we have trained the Bi-LSTM model on the classification task and then extracted motion features as the concatenation of hidden layers h_l and h'_1. In [43], we fix both the hidden layers to 512 dimensions, resulting in the output of $1,024$ dimensions. By adjusting the hidden state size we can simply control the trade-off between efficiency and descriptive power of extracted features. Compared

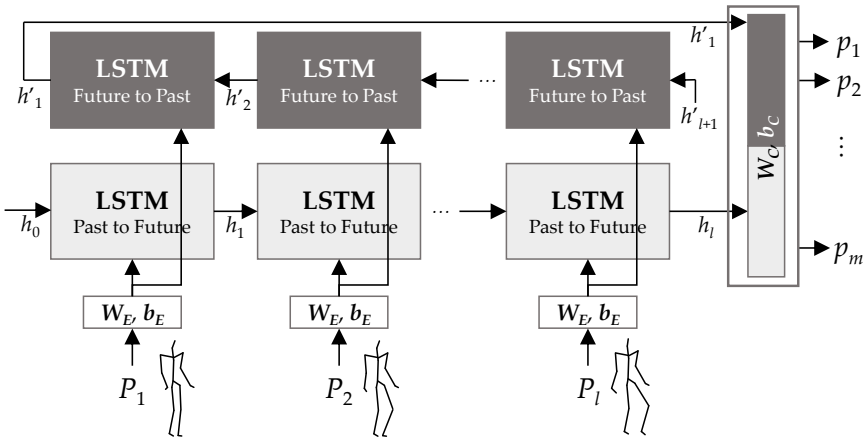


Figure 2.2: Schema of the Bi-LSTM architecture trained on the classification task [44] – individual action poses $P_1, \ldots, P_l$ are gradually embedded into LSTM cells in both past-to-future and future-to-past directions to determine classification probabilities $p_1, \ldots, p_m$ of m classes. The action feature can then be extracted by concatenating the hidden states h'_1 and h_l.

to CNNs in [34, 41] with the last hidden layer of $4,096$ dimensions, this approach keeps 4-times smaller features and can be directly trained using raw skeleton coordinates, instead of intermediate motion-image representations. The output features can be finally compared using the Manhattan or Euclidean distance functions, that achieve a comparable result quality.

2.1.3 Motion-Word Features

The proposed CNN or Bi-LSTM features are very effective for the action recognition task when labeled pre-segmented actions are provided. However, there is a growing amount of motion data captured as a *continuous* 3D skeleton sequence without any information about its semantic partitioning. To make such unsegmented and unlabeled data efficiently accessible, we have proposed to transform them into a structured text-like representation [37], to which mature text retrieval models could be possibly applied. Specifically, each long motion is synthetically partitioned into a sequence of short segments that are quantized into *motion words* (MWs) – compact features with similar characteristics as words in text documents. The similarity of variable-length motion-word sequences is determined using the DTW function.

The main issue here is to find an effective quantization of the motion segments to build a vocabulary of MWs. The most desirable MW property is that two MWs match each other if their corresponding segments exhibit similar movement characteristics, and do not match if the segments

are dissimilar. This is challenging with the quantization approach, since it is generally not possible to divide a given space in such way that all pairs of similar objects are in the same partition. Some pairs of similar segments thus get separated by partition borders and become non-matching. We deal with this problem by designing *soft* MWs that are more complex structures keeping information also about neighboring partitions. The soft MWs demonstrate much better ability to preserve the motion content, but their processing is more computationally demanding because of their non-trivial matching. More details about this approach can be found in the attached publication in Part II (Work 2).

In [38], we have also successfully applied the motion-word concept to medium-sized skeleton sequences, so-called *episodes*, taking from dozens of seconds to several minutes (e.g., a figure-skating performance). We have especially built a MW vocabulary for episode data and designed new matching functions by employing the advances known from the text-document processing. This has resulted in much more efficient similarity comparison with respect to the expensive DTW function used in [37]. This approach was selected into the best-paper session within the ISM 2020 conference, held online. More details about this approach can be found in the attached publication in Part II (Work 3).

2.2 Gait Recognition

Gait recognition is the problem of identifying people based on the way they walk. It is also one of the first application-oriented tasks that have tried to employ 3D human skeleton data [72]. Today, there are deep-learning approaches that strive to extract descriptive gait features from different human motion modalities captured by ordinary video cameras, floor sensors, radars, or accelerometers [22]. As the accuracy of gait recognition methods is disputable at larger scales, the gait modality can be used as a complementary approach in fingerprint- or face-recognition systems.

Although our main objective is general-purpose management of human skeleton data, our first attempts [39, 40] were designed and evaluated on the specific task of gait recognition. In the following, we briefly describe our initial idea – comparison of movement patterns on semantically meaningful parts that correspond to individual *gait cycles*, i.e., the left and right footstep. The cycles are then processed to extract their handcrafted gait features, whose similarity is quantified by a time warping function. The persons are finally recognized using a 1-nearest neighbor (1NN) classifier.

2.2.1 Walk Cycle Detection

We have proposed a specialized algorithm [39] to detect individual gait cycles within a long motion sequence. This algorithm firstly localizes all local minima within a time series representing how the distance between the left and right foot changes as the person walks. The segments between the consecutive minima correspond to individual footsteps. Since human walking might not be balanced, e.g., due to some injury, we also distinguish whether a given footstep is performed by the left or right leg by analyzing the additional time series between the left knee and right foot. As a proper gait cycle, we select a pair of the left footstep and consecutive right footstep. As natural variation in walking behavior results in different lengths of detected cycles, the gait cycles are finally normalized to a fixed length. For example, in [39] the cycles are normalized to a length of 150 video frames, which is the length of average cycle of the CMU dataset captured at the 120 frame-per-second rate.

2.2.2 Person Identification using Static and Dynamic Features

Person's walk can be characterized by a time series of distances between a pair of selected joints on the human body. In [40], we select four pairs of joints on legs and hands and extract the corresponding four time series for each detected gait cycle as *dynamic* features. We determine a similarity between two time series by the Dynamic Time Warping (DTW) function. We then estimate a similarity between two gait cycles by summing DTW similarities of that four pairs of time series. The experiments evaluated on the 131 walking sequences of 24 persons of the CMU dataset show the recognition rate of $96\,\%$ using the 1NN classifier.

To additionally increase recognition accuracy, we have extracted *static* skeleton features, represented by lengths of important bones on the human body, and fuse them with the dynamic features [73]. We have also proposed how lengths of specific bones can be better estimated in case the captured data exhibit some tracking errors [53], as in case of Kinect devices.

2.2.3 Prototype Implementation

The proposed gait-cycle detection and gait recognition algorithms are also available in the form of software [74], so-called MotionMatch software. This software encapsulates an application that demonstrates the gait recognition capabilities on the concatenation of the publicly available CMU and HDM05 datasets. To further increase the accuracy of person identification, we have considered the face modality and employed the MPEG-7 descriptor to recognize people based on their faces. In particular, we have developed a multi-modal software [73], so-called MMPI, that recognizes peo-

ple based on a weighted fusion of both the face and gait modalities. This software also includes a graphical user interface to demonstrate the multimodal recognition characteristics.

2.3 Action Recognition

Action recognition, also referred to as action classification, is probably the most popular motion-analysis operation [75, 76]. It is the problem of inferring the kind of a 3D skeleton action based on a labeled dataset of training actions. Solving this problem is difficult as the actions of the same kind, i.e., belonging to the same class, can be performed by various subjects in different styles, speeds, and initial body postures.

Currently, action recognition is almost exclusively solved by training a deep neural-network classifier that can effectively learn semantic relationships among related training actions. In deep learning, 3D skeleton actions are transformed into intermediate representations (e.g., graph structures [77, 69] or 2D motion images [27, 67]) that are used to train some kind of classification model, commonly based on convolutional neural networks (CNN) [66, 67], graph convolutional networks (GCN) [68, 69], Long Short-Term Memory (LSTM) networks [70, 71], or their combinations [78]. The trained models are then directly used for classification of input actions. Alternatively, the trained models can be used for extraction of high-dimensional deep features, as stated in Section 2.1. The features are compared by a distance measure to find the most similar training actions with respect to a query action being classified. The retrieved samples are then processed by a k-nearest-neighbor (kNN) classifier to determine the class of the query action [34]. Although kNN classifiers require additional processing costs, they do not need to be expensively retrained when new action classes appear, compared to standalone neural networks.

In the context of action recognition, we mainly focus on kNN classification. We have proposed several kNN classifiers that employ the 4,096D features extracted from a CNN. We have also proposed new action augmentation techniques and shown how they can significantly help to increase recognition accuracy in combination with the LSTM features. We present these achievements in the following.

2.3.1 *k*NN Classification

To compare a pair of actions, we adopt our similarity metric originally proposed in [34] and improved in [33]. In particular, we fine-tune the AlexNet convolutional network by motion images of training actions and extract the corresponding 4,096D deep feature vectors, which can be compared by the Euclidean distance to determine their similarity. By comparing the feature

vector of a query against the feature vectors of all training actions, we identify the k-most similar ones. In [34, 33], we have simply set k to 1 and classified the query based on the class of the nearest neighbor only. Although the 1NN classifier is simple and quite accurate, it need not be convenient when the query-closest neighbors have almost the same distance while belonging to different classes. To solve this problem, we have introduced a *weighted-distance* (WD) kNN classifier [41] that recognizes the query based on the combination of both class-assignment and similarity of the nearest neighbors.

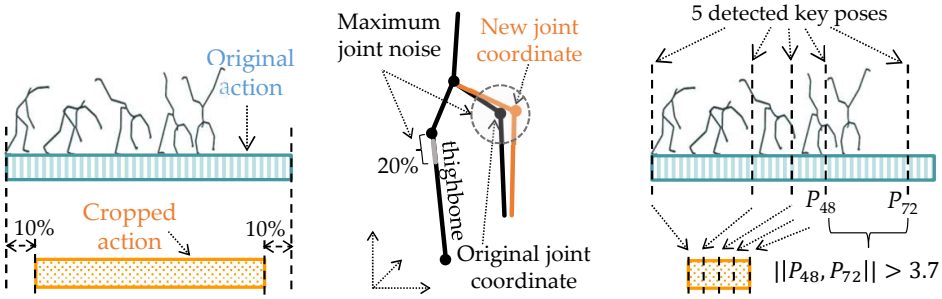
2.3.2 Confusion-based *k*NN Classification

Action recognition is difficult when the correct class is confusable with another class, e.g., "grab a thing" with "deposit a thing". In such cases, the WD classifier can return very similar probabilities for the two best matching classes. In [41], we have proposed to apply the WD classifier to identify the best matching classes and then *re-rank* the retrieved k-nearest neighbors based on different similarity measures which can better separate the top ranked classes, than the original measure [33]. We can define many handcrafted-based similarity measures and automatically select the most useful one for each pair of classes using *confusion matrices* learned from the training data. The class of the neighbor with the smallest re-ranked distance is finally considered as the classification result. More details about this approach can be found in the attached publication in Part II (Work 4).

2.3.3 Bi-LSTM Recognition with Data Augmentation

Current research in action recognition suggests to employ various architectures of deep neural networks [79]. However, the quality of such proposals much depends on the size of training datasets. It is not easy to train well-performing models using a small number of action samples in each class, as in case of the HDM05-130 dataset distinguishing in 130 classes but providing only about 10 class samples when $50\,\%$ of training data are used. Although providing new training data is feasible in other domains, e.g., in the image domain, it is much more difficult to obtain new high-quality samples of 3D skeleton sequences, mainly due to the high costs of motion capture technologies and an absence of professional actors. In [43, 44], we have proposed several augmentation techniques for the domain of 3D actions in order to artificially enlarge training data. As illustrated in Figure 2.3, the proposed techniques deform either the spatial, or temporal dimension of original actions, and thus contribute to higher intra-class variances compared to the original actions.

As a first step towards more accurate action recognition, we have generated various sets of augmented training data and used them to train the Bi-



(a) Crop augmentation: cropping the original action by 20 %, i.e., 10 % of the content is trimmed on the left and 10 % on the right side w.r.t. original action length.

(b) Noise augmentation: moving a joint into a new random position, which is at most 20 % of the thighbone length away from the original position.

(c) Key pose augmentation: five key poses are detected, where the distance between any two consecutive key poses is higher than the key pose threshold set to 3.7.

Figure 2.3: Illustration of three selected techniques for augmentation of 3D skeleton actions [44].

LSTM classifier [43]. As reported in the last row in Table 2.1, the achieved recognition rate outperforms the state-of-the-art results on the HDM05-130 dataset. Even though the Bi-LSTM classifier with augmented data performs very well, it is generally very hard to determine what are the most suitable augmentation techniques for a given dataset. Assume n augmentation techniques are available, then there are 2^n possible combinations how different sets of augmented training data can be generated. For example, if $n = 16$, there are $65,536$ different subsets of combinations. And it is not computationally feasible to train such number of Bi-LSTM classifiers for choosing the best combination for each dataset. To overcome this problem, we have proposed to (i) train only one independent classifier for each of the n augmentation techniques and (ii) estimate the accuracy of a specific combination by efficient fusion of the corresponding classification results of the independent classifiers. This has enabled us to fast estimate the suitability of augmentation techniques for the HDM05-130 dataset and helped slightly improve recognition accuracy. More details about the whole approach can be found in the attached publication in Part II (Work 5).

2.3.4 Prototype Implementation

We have also developed a prototype [42] that utilizes the weighted-distance 3NN classifier for recognizing actions represented by the 4,096D CNN features. This prototype application allows a user to browse long motion sequences and specify any subsequence as the input for probabilistic classi-

Table 2.1: Comparison of action-recognition accuracy with the state-of-the-art methods using the 2-fold cross validation (i.e. 50 % of training data) on the HDM05-130 ground truth. The methods are sorted by achieved accuracy.

Method[1)]	**Classifier**	**Accuracy**
[80] *(2017)*	LieNet-2Blocks	75.78 %
[67] *(2017)*	CNN on motion images	83.33 %
[28] *(2020)*	DMT-Net	85.30 %
[69] *(2019)*	Si-GCN	85.45 %
[81] *(2020)*	PGCN-TCA	86.59 %
*[34] *(2018)*	CNN features + 1NN	86.79 %
*[33] *(2017)*	Enh. CNN features + 1NN	87.38 %
[82] *(2018)*	PB-GCN	88.17 %
*[41] *(2018)*	CNN & handcrafted features + confusion kNN	88.78 %
*[43] *(2019)*	Bi-LSTM with augmented actions	92.92 %

[1)] Our proposed methods are denoted by the star symbol (*)

fication based on the 130 predefined HDM05-130 classes, as schematically illustrated in Figure 2.4.

2.4 Subsequence Search

The subsequence retrieval operation aims at inspecting long data sequences and detecting such their subsequences that are highly similar to a short query motion. This task is difficult as query-relevant subsequences can occur arbitrarily within the data sequences and can vary in lengths based on the speed of execution. The retrieval process can be generally divided into two steps: *search* and *refinement*. In the search step, a set of query-relevant *candidate* results is efficiently retrieved, e.g., using various index structures, such as the tries in [83] or M-index in [46]. In the refinement step, the retrieved candidates are re-ranked by more expensive techniques (e.g., PageRank in [84] or ranking by DTW in [85]) to determine the final results. When the search step is not supported [76, 86], the refinement is evaluated over the whole database.

The main problem in subsequence retrieval is to find the precise alignment of an arbitrary query within a long data sequence. This can be done by expensive time-warping functions that match the query and a data subsequence on the level of individual poses [86]. Alternatively, the proper alignment can be found by segment-level matching, which requires to partition the query [83] or data [51] motions into sequences of overlapping *segments* – short subsequences of poses. However, the search phase then

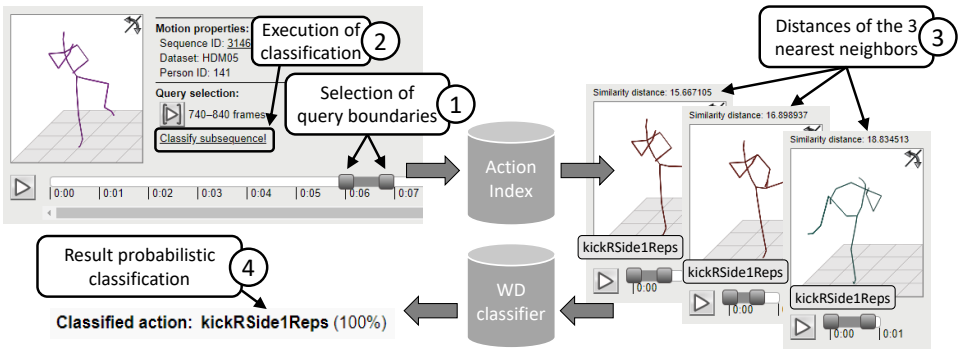


Figure 2.4: Schematic screenshot of the action-recognition application [42]. The input motion is selected as a subsequence between 740–840 frames (i.e., roughly between sixth and seventh second) and finally recognized as the "kickRSide1Reps" class with the 100 % probability, since all the three most similar retrieved actions correspond to that class.

becomes more difficult since a sequence of query segments has to be found within a long sequence of data segments. This is usually solved by temporal filters [83] or specialized functions, such as Earth-mover's distance (EMD) [76] or Longest Common Subsequence (LCS) [87], that operate over more-compact segment features.

In the following, we present our subsequence retrieval techniques that find proper alignment of a query within a data sequence based on matching individual poses [46, 47], or fixed-size segments [49, 48, 51].

2.4.1 Pose-Based Indexing

We have firstly proposed a retrieval algorithm [46] that indexes long motions on the level of individual poses, which are represented as handcrafted 28D feature vectors of angles between selected pairs of bones [47]. In the search phase, the algorithm selects *key poses* of a query motion and efficiently retrieves the candidate poses within a data motion which are similar to any query key pose. In the refinement phase, the temporal surroundings of the candidate poses are carefully examined to determine relevant subsequences. Such subsequences are then compared against the query using the Manhattan distance function and the most similar and non-overlapping ones are returned as the query result. More details about this approach can be found in the attached publication in Part II (Work 6).

2.4.2 Segment-Based Matching

However, indexing on the level of individual poses is not much convenient since the temporal dimension is ignored and so many poses can become the candidates for the refinement phase. In [49], we have proposed to take the temporal dimension into account by indexing long motion sequences on the level of short segments. In the pre-processing step, we partition input long motions into sequences of segments of about 1 second duration, extract their deep 4,096D CNN features, and index them. During query processing, we also partition the query into several segments, extract their deep features, and efficiently search for the candidate data segments that are the most similar to the query segments. The surrounding around each candidate segment is specifically inspected to locate relevant subsequences that are finally refined against the query based on the Euclidean distance between their deep features. More details about this approach can be found in the attached publication in Part II (Work 7).

2.4.3 Multi-Level Segment-Based Matching

We further simplify query processing by skipping the refinement phase, which can be very time-consuming (e.g., evaluation of the PageRank algorithm in [84]). By skipping the refinement, we need to retrieve query results directly in the search phase. Our key idea is to consider the whole query as a *single* segment [48]. However, this requires to partition the long data motions into segments of lengths that are similar to any future query length. As the length of future queries can be lower- and upper-bounded in advance, we have proposed to partition the data motions multiple times into sequences of segments of different lengths, that are organized within a multi-level segmentation structure – see Figure 2.5 for more details. In the search phase, the query-similar subsequences can then be easily and efficiently located by searching just a single level of segments – whose length is the closest to the query length – without any need of additional expensive post-processing. This approach was selected among the best five papers within the SISAP 2016 conference, held in Tokyo, Japan. We have further extended this idea in [51] where several segmentation levels are searched in parallel to locate also subsequences that are performed more slowly and faster with respect to the query performance speed. More details about the whole multi-level and speed-invariant approach can be found in the attached publication in Part II (Work 8).

In Table 2.2, a comparative summary of our and existing subsequence search methods is provided from several views: (i) the expansion of query into overlapping segments, which leads to several sub-queries that have to be independently evaluated and their results post-processed, (ii) the volume of data replication caused by partitioning data sequences into over-

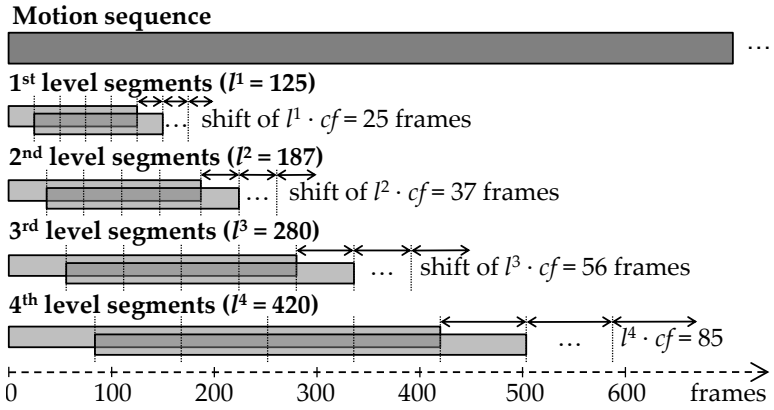


Figure 2.5: Four-level segmentation structure built over a single skeleton sequence [48]. This structure is sufficiently dense to evaluate any query whose length is bounded from 100 to 500 frames (i.e., query duration between 0.8–4.2 seconds in case of the 120 FPS rate). The i-th level contains segments of the same length l^i that are shifted by $cf \cdot l^i$ frames, where $cf = 0.2$ stands for the segmentation density factor.

lapping segments, (iii) the usage of the search step, (iv) or the way of evaluation of the refinement step. The table also provides query response time (QRT) that corresponds to the actual time needed to answer a single query. Such query response times cannot be directly compared, as they are taken from published papers and much depend on several factors, such as the database size, the data frame-per-second rate, selected queries, or used hardware. It seems that the approach in [83] reports the most promising performance, however, troubles arise in the case when different queries are evaluated and possibly all branches of the used trie structure must be visited to locate the beginning (or end) of the query. Moreover, search quality is limited due to the adopted handcrafted features. By contrast, our approach in [51] incorporates very effective deep features that can be further indexed by some metric-based index structure. This could possibly enable to search online in large skeleton-data volumes of weeks or even months.

2.4.4 Prototype Implementation and Demonstration Application

We have developed an online web application [50] that allows users to evaluate subsequence-search queries in the two popular motion capture datasets: HDM05 and CMU. These datasets contain 324 and $2,191$ motion sequences with the average length of $4,699$ and $1,750$ frames, respectively. The total length of all $2,515$ sequences is more than 5.3 millions of frames, which corresponds to about 12 hours with the sampling frequency of 120 Hz.

Table 2.2: Methods for subsequence searching in long motions.

Method[1]	Features	Data rep.[2]	Query expan.	Search	Refinement	Efficiency QRT	Efficiency DB size
[85] *(2005)*	geom. relations	○	–	linear scan	DTW	294 ms	3 h
[86] *(2006)*	geom. relations	○	–	–	DTW	10^4 ms	0.5 h
[88] *(2009)*	motion patterns	○	✓	–	body-part fusion	72 ms	4 h
[87] *(2011)*	joint rotations	○	–	–	LCS	10^5 ms	9 h
*[46] *(2013)*	joint rotations	○	✓	M-index	temporal filter	10^3 ms	1 h
[83] *(2013)*	geom. relations	○	✓	trie	temporal filter	40 ms	35 h
*[49] *(2017)*	deep features	◐	✓	linear scan	Euclidean	10^3 ms	1 h
[76] *(2018)*	deep motifs	◐	–	–	EMD	10^{3-4} ms	0.25 h
*[51] *(2019)*	deep features	●	–	–	Euclidean	84 ms	1 h
[84] *(2019)*	3D coordinates	○	–	text search	PageRank	40 ms	9 h

[1] Our proposed methods are denoted by the star symbol (*)

[2] Data replication – none (○), overlapping data segments (◐), overlapping data segments organized in multiple levels (●)

The proposed application does not require any textual annotations nor explicit knowledge of the data and can deal with spatio-temporal variances of human movements. It is effective due to the integration of deep features reaching high-quality results in action recognition [34]. It is also very efficient by locating query-similar subsequences in the 12-hour motion database in less than 1 s. A live demo of subsequence search is running publicly available at: `http://disa.fi.muni.cz/mocap-demo/`.

2.5 Action Detection

Action detection, sometimes referred to as annotation, is the problem of identifying actions within a long skeleton sequence. The actions can also be detected within a continuous stream, which typically requires real-time processing. In contrast to the subsequence search operation, the examples of labeled training actions are provided in advance. Such training actions can be pre-processed to extract their deep features [52] or to build motion templates [89, 64] that aggregate the actions of the same class into a single representation. The pre-processed actions or templates are then matched with the input skeleton sequence/stream to detect the desired actions. The actions are matched either on the level of individual *poses* (i.e., frames), or artificial *segments* that are gradually extracted from the streaming input.

The segment-level approach uses a sliding window to partition the sequence either artificially into many overlapping segments [64, 90, 91], or semantically into disjoint segments [92, 93, 94]. The obtained segments can then be directly recognized using a learned classifier [94], or matched against the pre-processed templates or actions based on different distance functions, such as the Euclidean distance in [52], DTW in [64], or fusion of

Table 2.3: Methods for action detection in long motions or streams.

Method[1)]	Temporal mechanism	Type	Early det.	Predict.	FPS speed
[64] *(2009)*	DTW + motion templates	segment	–	–	240
[94] *(2017)*	Naive Bayes + Riemannian manifold	segment	–	–	7
*[52] *(2017)*	kNN classifier + CNN features	segment	–	–	131
[92] *(2018)*	Curvilinear seg. + fusion of classifiers	segment	✓	–	667
[93] *(2019)*	LSTM + sliding window features	segment	–	–	5.4
[95] *(2015)*	SVM + temporal pyramids	pose	–	–	380
[91] *(2016)*	Linear search + BoG + sliding window	pose	✓	✓	93
[32] *(2016)*	Classification-regression LSTM	pose	✓	✓	1,230
[89] *(2018)*	Linear search + bag of gestures (BoG)	pose	✓	–	N/A
[30] *(2018)*	Attention-based LSTM	pose	✓	–	N/A
* [31] *(2019)*	Online-LSTM	pose	✓	✓	7,700

[1)] Our proposed methods are denoted by the star symbol (*)

linear classifiers in [92]. The segment is finally annotated by the label of the best-matching template or action if the distance satisfies some threshold condition.

The pose-level approach typically uses LSTM networks [31, 30, 32] or Support Vector Machines [95] to learn a classification model that estimates a class-relevance probability for each pose of the streaming input. To consider the neighboring context of individual poses, the recent past is encoded within enriched pose features [62] or within the memory of LSTM networks [30]. The advantage of the pose-level approach is that it can discover actions before they finish [96, 62] (so-called *early detection*), or even *predict* future actions [91, 32].

Nevertheless, a general disadvantage of all the detectors which internally use a neural network for classification of segment or pose data into the predefined number of classes, is that they need to be completely retrained whenever a new class of actions is introduced [4]. The pose-level and segment-level action detection methods are summarized in Table 2.3. In the following, we present our action detectors based on both segment-level [52] and pose-level [31] matching.

2.5.1 Segment-Based Action Detection

Similarly as in our subsequence-search paper [48], we gradually build the multi-level segmentation structure over an input motion sequence to be annotated [52]. The number of levels and corresponding lengths of artificial segments are determined based on the lengths of training actions. During the annotation phase, each artificial segment is processed by extracting its deep 4,096D CNN feature that is compared against the deep features of training actions. If the similarity of the nearest neighbor satisfies the threshold condition, the segment is assigned the neighbor class, i.e., all the

poses covered by that segment get such class label. This approach was awarded as the Best Student Paper at the IEEE ISM 2017 conference, held in Taichung, Taiwan. More details about this approach can be found in the attached publication in Part II (Work 9).

2.5.2 Pose-Based Action Detection

Segment-based annotation generally suffers from (i) a large number of similarity comparisons between the segments and training actions, (ii) not precise marking of the beginnings and endings of detected actions, and (iii) necessity of reading each segment before its processing, implying that annotations are discovered with a slight delay. We have suppressed these disadvantages by proposing two pose-based annotation algorithms [31] that are based on the LSTM and Bi-LSTM neural networks. Such networks have already proven to be successful in recognizing pre-segmented actions [71, 43]. In particular, we have proposed an online action detection algorithm (Online-LSTM) able to recognize precise beginnings and endings of concurrent actions within skeleton streams. We have shown that the beginnings of actions are detected immediately, without the necessity to wait for their termination, which enables predicting actions a few hundreds milliseconds ahead. Additionally, we have proposed an offline algorithm (Offline-LSTM) that utilizes a bidirectional LSTM network to further enhance annotation accuracy by analyzing also the future-to-past context. This limits the Offline-LSTM algorithm to be applied to streams, as the whole sequence needs to be available in advance. In contrast to standard algorithms, both approaches provide a multi-label annotation of actions that can be performed concurrently. The results on the long skeleton sequences of the HDM05 dataset outperform the state-of-the-art approaches not only in effectiveness, but also in efficiency, as our approach is at least one order of magnitude faster, capable of annotating roughly 10 k poses per second. More details about this approach can be found in the attached publication in Part II (Work 10).

Chapter 3

Conclusions and Future Research Directions

The last decade's research has established many fundamental techniques for content-based similarity management of 3D human skeleton data, especially from the perspective of the action recognition, action detection, and subsequence search tasks. In the context of such tasks, this thesis briefly summarizes the state-of-the-art principles and compares them to the achievements published in the papers in which Jan Sedmidubský participated as the co-author. Recently, we have decided to extend such state-of-the-art comparison along with outlined future challenges into the form of a survey paper [4], which is currently under review. The big interest in the topic of human-motion processing is supported by our tutorials accepted to computer-science conferences (ACM Multimedia and ACM ICMR) and by the invited seminar lecture we had within the medicine conference (ESMAC). This topic is also highly interdisciplinary which is supported by diverse motion-processing papers appearing in various domains, such as computer science, sports, or medicine. From the computer-science point of view, a large number of related papers also appear in different fields, such as computer vision, multimedia, or information retrieval.

The current situation in skeleton data management can be generally characterized in a way that there are content-based processing technologies operating over relatively *small* and *single-person* collections. However, the current progress in technologies and pose-estimation software tools [97, 1, 98, 99, 100] suggests that massive volumes of 2D (or even 3D) skeleton data will soon be available from ordinary cameras or videos uploaded and freely available on the web. Apart from being voluminous, such motion data are likely to be imprecise due to constrained video resolution, limited accuracy of the pose-estimation methods, reduced frequency of frame rates, or occlusions. At the same time, the video-based data will often contain multiple, possibly interacting, entities (e.g., individuals or groups). In

general, we expect the gradual shift in research focus from *single-person, small, precise*, and *uni-modal* data collections to *groups of people, huge, imprecise*, and *multi-modal* data collections.

We believe that this paradigm shift offers unique research opportunities. Therefore, we outline and discuss several types of challenges brought by the expected nature of future skeleton data and technologies in the following. We first focus on the applicability of existing techniques to the massively produced data and discuss the issues related to data cleaning, metric learning, and searching. Then, we take a step beyond the established areas of action recognition, action detection, and subsequence search and outline new possibilities for analyzing the motion data content from the perspective of complex queries and group understanding.

Data Cleaning

The extraction of 2D or 3D skeleton data from ordinary videos is likely to produce datasets of uncertain quality that will need to be cleaned and enhanced. Though there are some techniques on motion data cleaning, they mainly focus on correcting small data errors coming from marker-based capturing systems using statistical methods, which are not applicable to highly erroneous video-based 2D skeleton sequences [101]. This requires to study alternative data-cleaning directions, for example, to enhance the estimation of imprecise joint coordinates by additional visual modalities such as colors, faces, or context in general, which can also be extracted from the video data [102].

Crawling web videos and extracting the corresponding skeleton data also brings the need to detect duplicate and near-duplicate motion sequences. In the field of general content-based retrieval, the similarity join operator [103] is used to detect very similar objects. By adapting this operator to the motion processing domain, all pairs of crawled skeleton sequences within a certain similarity threshold could be efficiently located and further analyzed to reveal the duplicates.

Metric Learning

Most of the existing similarity metrics are learned using various kinds of deep neural networks in a supervised way, by providing a rather low number of application-specific motion classes for which high-quality training data exist. Nevertheless, the usability of the learned metrics to 2D skeleton data, new application domains, or larger datasets is limited by the availability of training data and the ability of the deep neural networks to deal with a growing number of classes, which has not been much studied yet. In this respect, there are three important research directions that should

be considered. First, new reference collections of cleaned, precise, and labeled data should be built for supervised metric learning as well as for evaluating benchmarks. Compared to the current situations when training data are often created and labeled manually, the building of large future collections could be done in a crowdsourcing manner, e.g., using crowdsourcing, relevance feedback, or gamification [104]. Second, in addition to the skeleton data, other visual modalities could be extracted from the video data and used to better distinguish among the growing number of motion classes [105]. The utilization of orthogonal modalities should be especially useful in situations when reliable training data are not available. Third, in environments where labeled training data are difficult to obtain, unsupervised learning approaches could be adopted, such as the triplet-loss learning that requires to provide the examples of similar and dissimilar motions with respect to the training ones [76]. Such examples could be obtained by crowdsourcing or by defining some coarse-grained matching function capable of recognizing only between similar and dissimilar motions.

To be able to integrate a learned metric into large-scale retrieval systems, it is important that the learned motion features as well as the corresponding comparison function can be efficiently indexed. Since the state-of-the-art features are typically extracted from the hidden layers of deep neural networks in the form of high-dimensional vectors, their indexing becomes problematic due to the curse of dimensionality. Therefore, it is challenging to propose indexable motion features that would provide a reasonable trade-off between their descriptiveness and complexity [37].

Scalable Searching

The state-of-the-art retrieval techniques are primarily designed to operate on 3D skeleton-data collections of the maximum length of dozens of hours. For collections of 2D skeleton sequences which are considered several orders of magnitude larger, we need completely new and scalable algorithms for both search and subsequence search operations. In contrast to the current skeleton-data retrieval techniques with linear [76] or sub-linear search complexity [46], there is a need to develop approximate search strategies with nearly constant processing costs while reaching reasonable quality of the query results. One of the possible solutions could be to apply some content-preserving transformation of 2D skeleton sequences into structured text-like documents and index such documents based on adapted text-based processing principles, which are successfully used by large-scale text search engines [106]. Another possible approach could transform 2D skeleton data into compact fixed-size bit representations [107] and employ the efficient Hamming distance to compare a pair of motions. To efficiently access the most query-relevant motions, the bit representations could be indexed by generic metric-based structures [108].

Evaluating Complex Queries

Current research mainly focuses on processing short motions with a clear semantics, e.g., recognizing the classes of short actions, detecting short actions in long motions, or searching for the sub-motions that are the most similar to a short query. On the other hand, there are application scenarios where more complex motion sequences and their relationships need to be analyzed. Let us again consider the figure-skating scenario: we might be interested in performances with two triple-jumps at the beginning and a five-second spin towards the end. This shifts the focus of queries from short actions to complex recordings that consist of multiple actions while representing some real-world semantic unit (e.g., a figure-skating performance) [38]. Evaluation of such types of complex queries requires a complete re-thinking of skeleton-data management techniques that currently focus on evaluation of standard k-nearest neighbor queries. A possible solution could (i) decompose a query into many segments, (ii) search for query-relevant data segments using standard techniques, (iii) compose the retrieved segments into candidate sequences while respecting the segment sequentiality with respect to the query, and (iv) refining the constructed sequences based on additional query requirements.

Processing of Multi-Subject Recordings

Understanding behavior of groups of people is highly desirable in many domains, such as smart cities, psychology, or human-computer interaction. However, existing methods for motion understanding typically consider only single-person recordings. Interactions among more subjects are studied rarely [24], and usually involve only activities of pairs [90] in specific application scenarios. The research challenge is to propose generic approaches for matching similar movement patterns within multiple interacting subjects. A big potential especially lies in designing methods able to determine the similarity of performing activities of two groups containing a different number of subjects. This problem opens many research opportunities, for example, (i) recognition of group activities that are invariant to the number of subjects, (ii) detection of a subgroup of individuals performing a given activity, (iii) searching, eventually subsequence searching, in multi-person skeleton sequences where both database and query data can contain a different number of interacting subjects, (iv) discovering similar movement patterns in small groups, or (v) identifying semantically-related groups of individuals (e.g., families, couples, or friends) within crowded scenes. This would require to completely redefine the existing similarity models as well as the techniques for action recognition, action detection, and subsequence search, originally developed for single-subject motion recordings. To be able to evaluate future technologies, there is also a need

to collect new datasets and benchmarks for multi-subject processing. In addition, the standard query-by-example search paradigm could be substituted by alternative query construction approaches, as the example query might not simply exist due to the explosion of possibilities how multiple subjects can interact.

Bibliography

[1] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: real-time multi-person 3d motion capture with a single RGB camera," *ACM Transactions on Graphics*, vol. 39, no. 4, 2020.

[2] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3810–3818, IEEE Computer Society, 2015.

[3] R. Lun and W. Zhao, "A survey of applications and human motion recognition with Microsoft Kinect," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 5, pp. 1–48, 2015.

[4] J. Sedmidubsky, P. Elias, P. Budikova, and P. Zezula, "Content-based management of human motion data: Survey and challenges," *under review (2021)*, pp. 1–15.

[5] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 138:1–138:11, 2016.

[6] S. Starke, Y. Zhao, T. Komura, and K. Zaman, "Local motion phases for learning multi-contact character movements," *ACM Transactions on Graphics*, vol. 39, no. 4, 2020.

[7] M. J. Kyan, G. Sun, H. Li, L. Zhong, P. Muneesawang, N. Dong, B. Elder, and L. Guan, "An approach to ballet dance training through ms kinect and visualization in a cave virtual reality environment," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 2, pp. 23:1–23:37, 2015.

[8] F. Anderson, T. Grossman, J. Matejka, and G. W. Fitzmaurice, "Youmove: enhancing movement training with an augmented reality mirror," in *26th ACM Symposium on User Interface Software and Technology (UIST)*, pp. 311–320, ACM, 2013.

[9] Y. Yan, O. M. Omisore, Y. Xue, H. Li, Q. Liu, Z. Nie, J. Fan, and L. Wang, "Classification of neurodegenerative diseases via topological motion analysis - A comparison study for multiple gait fluctuations," *IEEE Access*, vol. 8, pp. 96363–96377, 2020.

[10] W. R. Johnson, A. Mian, C. J. Donnelly, D. G. Lloyd, and J. A. Alderson, "Predicting athlete ground reaction forces and moments from motion capture," *Medical and Biological Engineering and Computing*, vol. 56, no. 10, pp. 1781–1792, 2018.

[11] Y. Zhang and Y. Ma, "Application of supervised machine learning algorithms in the classification of sagittal gait patterns of cerebral palsy children with spastic diplegia," *Computers in Biology and Medicine*, vol. 106, pp. 33–39, 2019.

[12] R. Watari, D. Kobsar, A. Phinyomark, S. Osis, and R. Ferber, "Determination of patellofemoral pain sub-groups and development of a method for predicting treatment outcome using running gait kinematics," *Clinical Biomechanics*, vol. 38, pp. 13–21, 2016.

[13] L. Pogrzeba, T. Neumann, M. Wacker, and B. Jung, "Analysis and quantification of repetitive motion in long-term rehabilitation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 3, pp. 1075–1085, 2019.

[14] S. Schez-Sobrino, D. Vallejo-Fernandez, D. N. Monekosso, C. Glez-Morcillo, and P. Remagnino, "A distributed gamified system based on automatic assessment of physical exercises to promote remote physical rehabilitation," *IEEE Access*, vol. 8, pp. 91424–91434, 2020.

[15] C. Joyce, A. Burnett, J. Cochrane, and K. Ball, "Three-dimensional trunk kinematics in golf: between-club differences and relationships to clubhead speed," *Sports Biomechanics*, vol. 12, no. 2, pp. 108–120, 2013.

[16] A. Aristidou, A. Shamir, and Y. Chrysanthou, "Digital dance ethnography: Organizing large dance collections," *Journal on Computing and Cultural Heritage*, vol. 12, no. 4, 2019.

[17] H. Pirsiavash, C. Vondrick, and A. Torralba, "Assessing the quality of actions," in *11th European Conference on Computer Vision (ECCV)*, (Cham), pp. 556–571, Springer International Publishing, 2014.

[18] W. M. R. Wan Idris, A. Rafi, A. Bidin, A. A. Jamal, and S. A. Fadzli, "A systematic survey of martial art using motion capture technologies: the importance of extrinsic feedback," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 10113–10140, 2019.

[19] T. Shimizu, R. Hachiuma, H. Saito, T. Yoshikawa, and C. Lee, "Prediction of future shot direction using pose and position of tennis player," in *2nd International Workshop on Multimedia Content Analysis in Sports (MMSports)*, (New York, NY, USA), pp. 59–66, ACM, 2019.

[20] D. Zecha, C. Eggert, and R. Lienhart, "Pose estimation for deriving kinematic parameters of competitive swimmers," *Electronic Imaging*, pp. 21–29, 2017.

[21] M. Einfalt, C. Dampeyrou, D. Zecha, and R. Lienhart, "Frame-level event detection in athletics videos with pose-based convolutional sequence networks," in *2nd International Workshop on Multimedia Content Analysis in Sports (MMSports)*, pp. 42–50, ACM, 2019.

[22] C. Wan, L. Wang, and V. V. Phoha, "A survey on gait recognition," *ACM Computing Surveys*, vol. 51, no. 5, pp. 89:1–35, 2018.

[23] M. M. Islam, A. Lam, H. Fukuda, Y. Kobayashi, and Y. Kuno, "An intelligent shopping support robot: understanding shopping behavior from 2d skeleton data using gru network," *ROBOMECH Journal*, vol. 6, no. 1, p. 18, 2019.

[24] T. Hu, X. Zhu, S. Wang, and L. Duan, "Human interaction recognition using spatial-temporal salient feature," *Multim. Tools Appl.*, vol. 78, no. 20, pp. 28715–28735, 2019.

[25] P. Woznowski, A. Burrows, T. Diethe, X. Fafoutis, J. Hall, S. Hannuna, M. Camplani, N. Twomey, M. Kozlowski, B. Tan, N. Zhu, A. Elsts, A. Vafeas, A. Paiement, L. Tao, M. Mirmehdi, T. Burghardt, D. Damen, P. Flach, R. Piechocki, I. Craddock, and G. Oikonomou, *SPHERE: A Sensor Platform for Healthcare in a Residential Environment*, pp. 315–333. Springer, 2017.

[26] E. Barsoum, J. Kender, and Z. Liu, "HP-GAN: probabilistic 3d human motion prediction via GAN," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1418–1427, IEEE Computer Society, 2018.

[27] Q. Ke, M. Bennamoun, H. Rahmani, S. An, F. Sohel, and F. Boussaid, "Learning latent global network for skeleton-based action prediction," *IEEE Transactions on Image Processing*, vol. 29, pp. 959–970, 2020.

[28] T. Zhang, W. Zheng, Z. Cui, Y. Zong, C. Li, X. Zhou, and J. Yang, "Deep manifold-to-manifold transforming network for skeleton-based action recognition," *IEEE Transactions on Multimedia*, pp. 1–12, 2020.

[29] W. Zheng, L. Li, Z. Zhang, Y. Huang, and L. Wang, "Relational network for skeleton-based action recognition," in *International Conference on Multimedia and Expo (ICME)*, pp. 826–831, IEEE, 2019.

[30] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3d action recognition and detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3459–3471, 2018.

[31] F. Carrara, P. Elias, J. Sedmidubsky, and P. Zezula, "Lstm-based real-time action detection and prediction in human motion streams," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27309–27331, 2019.

[32] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *14th European Conference on Computer Vision (ECCV)*, pp. 203–220, Springer, 2016.

[33] J. Sedmidubsky, P. Elias, and P. Zezula, "Enhancing effectiveness of descriptors for searching and recognition in motion capture data," in *19th International Symposium on Multimedia*, pp. 240–243, IEEE Computer Society, 2017.

[34] J. Sedmidubsky, P. Elias, and P. Zezula, "Effective and efficient similarity searching in motion capture data," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12073–12094, 2018.

[35] J. Valcik, J. Sedmidubsky, and P. Zezula, "Assessing similarity models for human-motion retrieval applications," *Computer Animation and Virtual Worlds*, vol. 27, no. 5, pp. 484–500, 2016.

[36] P. Elias, J. Sedmidubsky, and P. Zezula, "Motion images: An effective representation of motion capture data for similarity search," in *8th International Conference on Similarity Search and Applications (SISAP)*, (Cham), pp. 250–255, Springer International Publishing, 2015.

[37] J. Sedmidubsky, P. Budikova, V. Dohnal, and P. Zezula, "Motion words: A text-like representation of 3d skeleton sequences," in *42nd European Conference on Information Retrieval (ECIR)*, pp. 527–541, Springer, 2020.

[38] P. Budikova, J. Sedmidubsky, J. Horvath, and P. Zezula, "Towards scalable retrieval of human motion episodes," in *IEEE International Symposium on Multimedia (ISM)*, pp. 49–56, IEEE Computer Society, 2020.

[39] J. Valcik, J. Sedmidubsky, M. Balazia, and P. Zezula, "Identifying Walk Cycles for Human Recognition," in *Pacific Asia Workshop on*

Intelligence and Security Informatics (PAISI), pp. 127–135, Springer-Verlag, 2012.

[40] J. Sedmidubsky, J. Valcik, M. Balazia, and P. Zezula, "Gait Recognition Based on Normalized Walk Cycles," in *8th International Symposium on Visual Computing (ISVC)*, pp. 11–20, Springer, 2012.

[41] J. Sedmidubsky and P. Zezula, "Probabilistic classification of skeleton sequences," in *29th International Conference on Database and Expert Systems Applications (DEXA)*, (Cham), pp. 50–65, Springer International Publishing, 2018.

[42] J. Sedmidubsky and P. Zezula, "Recognizing user-defined subsequences in human motion data," in *International Conference on Multimedia Retrieval (ICMR)*, pp. 395–398, ACM, 2019.

[43] J. Sedmidubsky and P. Zezula, "Augmenting Spatio-Temporal Human Motion Data for Effective 3D Action Recognition," in *21st IEEE International Symposium on Multimedia (ISM)*, pp. 204–207, IEEE Computer Society, 2019.

[44] J. Sedmidubsky and P. Zezula, "Efficient combination of classifiers for 3d action recognition," *Multimedia Systems*, pp. 1–12, 2021.

[45] M. Balazia, J. Sedmidubsky, and P. Zezula, "Semantically consistent human motion segmentation," in *25th International Conference on Database and Expert Systems Applications (DEXA)*, (Cham), pp. 423–437, Springer, 2014.

[46] J. Sedmidubsky, J. Valcik, and P. Zezula, "A Key-Pose Similarity Algorithm for Motion Data Retrieval," in *15th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, (Berlin, Heidelberg), pp. 669–681, Springer, 2013.

[47] J. Sedmidubsky and J. Valcik, "Retrieving Similar Movements in Motion Capture Data," in *6th International Conference on Similarity Search and Applications (SISAP)*, pp. 325–330, Springer, 2013.

[48] J. Sedmidubsky, P. Elias, and P. Zezula, "Similarity searching in long sequences of motion capture data," in *9th International Conference on Similarity Search and Applications (SISAP)*, (Cham), pp. 271–285, Springer International Publishing, 2016.

[49] J. Sedmidubsky, P. Zezula, and J. Svec, "Fast subsequence matching in motion capture data," in *21st European Conference on Advances in Databases and Information Systems (ADBIS)*, (Cham), pp. 50–72, Springer International Publishing, 2017.

[50] J. Sedmidubsky and P. Zezula, "A web application for subsequence matching in 3d human motion data," in *19th International Symposium on Multimedia*, pp. 372–373, IEEE Computer Society, 2017.

[51] J. Sedmidubsky, P. Elias, and P. Zezula, "Searching for variable-speed motions in long sequences of motion capture data," *Information Systems*, vol. 80, pp. 148–158, 2019.

[52] P. Elias, J. Sedmidubsky, and P. Zezula, "A real-time annotation of motion data streams," in *19th International Symposium on Multimedia*, pp. 154–161, IEEE Computer Society, 2017.

[53] J. Valcik, J. Sedmidubsky, and P. Zezula, "Improving kinect-skeleton estimation," in *16th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pp. 575–587, Springer, 2015.

[54] J. Sedmidubsky, P. Elias, and P. Zezula, "Benchmarking search and annotation in continuous human skeleton sequences," in *International Conference on Multimedia Retrieval (ICMR)*, pp. 38–42, ACM, 2019.

[55] P. Elias, J. Sedmidubsky, and P. Zezula, "Understanding the Gap between 2D and 3D Skeleton-Based Action Recognition," in *21st IEEE International Symposium on Multimedia (ISM)*, pp. 192–195, IEEE Computer Society, 2019.

[56] P. Elias, J. Sedmidubsky, and P. Zezula, "Understanding the limits of 2d skeletons for action recognition," *Multimedia Systems*, pp. 1–15, 2021.

[57] J. Sedmidubsky and P. Zezula, "Similarity-based processing of motion capture data," in *Proceedings of the 26th ACM International Conference on Multimedia (MM)*, (New York, NY, USA), pp. 2087–2089, ACM, 2018.

[58] J. Sedmidubsky and P. Zezula, "Similarity search in 3d human motion data," in *International Conference on Multimedia Retrieval (ICMR)*, pp. 5–6, ACM, 2019.

[59] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation Mocap Database HDM05," Tech. Rep. CG-2007-2, Universität Bonn, 2007.

[60] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *International Conference on Computer Vision (ICCV)*, pp. 2248–2255, IEEE, 2013.

[61] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for skeleton-based human action understanding," in *Workshop on Visual Analysis in Smart and Connected Communities (VSCC)*, pp. 1–8, ACM, 2017.

[62] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online human gesture recognition from motion data streams," in *ACM Conference on Multimedia*, pp. 23–32, ACM, 2013.

[63] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, SCA 2011, pp. 147–156, ACM, 2011.

[64] M. Müller, A. Baak, and H.-P. Seidel, "Efficient and Robust Annotation of Motion Capture Data," in *ACM SIGGRAPH Eurographics Symposium on Computer Animation (SCA)*, pp. 17–26, ACM, 2009.

[65] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[66] Z. Ahmad and N. M. Khan, "Towards improved human action recognition using convolutional neural networks and multimodal fusion of depth and inertial sensor data," in *20th International Symposium on Multimedia (ISM)*, pp. 223–230, IEEE, 2018.

[67] S. Laraba, M. Brahimi, J. Tilmanne, and T. Dutoit, "3d skeleton-based action recognition by representing motion capture sequences as 2d-rgb images," *Computer Animation and Virtual Worlds*, vol. 28, no. 3-4, p. e1782, 2017.

[68] K. Liu, L. Gao, N. M. Khan, L. Qi, and L. Guan, "Graph convolutional networks-hidden conditional random field model for skeleton-based action recognition," in *21st International Symposium on Multimedia (ISM)*, pp. 25–31, IEEE, 2019.

[69] R. Liu, C. Xu, T. Zhang, W. Zhao, Z. Cui, and J. Yang, "Si-gcn: Structure-induced graph convolution network for skeleton-based action recognition," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[70] Y. Wu, L. Wei, and Y. Duan, "Deep spatiotemporal LSTM network with temporal pattern feature for 3d human action recognition," *Computational Intelligence*, vol. 35, no. 3, pp. 535–554, 2019.

[71] J. Liu, G. Wang, L. Duan, P. Hu, and A. C. Kot, "Skeleton based human action recognition with global context-aware attention LSTM networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2018.

[72] R. Tanawongsuwan and A. F. Bobick, "Gait recognition from time-normalized joint-angle trajectories in the walking plane," *International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. C, pp. II:726–II:731, 2001.

[73] J. Sedmidubsky, J. Valcik, and P. Zezula, "Multi-modal person identification," 2015. Software (http://disa.fi.muni.cz/demo/person-identification/).

[74] J. Sedmidubsky, J. Valcik, and P. Zezula, "Motion-match: Motion recognition technology," 2014. Software (http://disa.fi.muni.cz/motionmatch/).

[75] J. Zhu, W. Zou, Z. Zhu, and Y. Hu, "Convolutional relation network for skeleton-based action recognition," *Neurocomputing*, vol. 370, pp. 109–117, 2019.

[76] A. Aristidou, D. Cohen-Or, J. K. Hodgins, Y. Chrysanthou, and A. Shamir, "Deep motifs and motion signatures," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 187:1–187:13, 2018.

[77] R. Zhao, K. Wang, H. Su, and Q. Ji, "Bayesian graph convolution lstm for skeleton based action recognition," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 6882–6892, IEEE, 2019.

[78] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018.

[79] B. Ren, M. Liu, R. Ding, and H. Liu, "A survey on 3d skeleton-based action recognition using learning method," 2020.

[80] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1243–1252, IEEE, 2017.

[81] H. Yang, Y. Gu, J. Zhu, K. Hu, and X. Zhang, "PGCN-TCA: pseudo graph convolutional network with temporal and channel-wise attention for skeleton-based action recognition," *IEEE Access*, vol. 8, pp. 10040–10047, 2020.

[82] K. C. Thakkar and P. J. Narayanan, "Part-based graph convolutional network for action recognition," in *British Machine Vision Conference (BMVC)*, pp. 1–13, BMVA Press, 2018.

[83] M. Kapadia, I. Chiang, T. Thomas, N. Badler, and J. T. K. Jr., "Efficient motion retrieval in large motion databases," in *Symposium on Interactive 3D Graphics and Games (I3D)*, pp. 19–28, ACM, 2013.

[84] M. G. Choi and T. Kwon, "Motion rank: applying page rank to motion data search," *Vis. Comput.*, vol. 35, no. 2, pp. 289–300, 2019.

[85] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 677–685, 2005.

[86] M. Müller and T.Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pp. 137–146, Eurographics Assoc., 2006.

[87] C. Ren, X. Lei, and G. Zhang, "Motion data retrieval from very large motion databases," in *International Conference on Virtual Reality and Visualization*, pp. 70–77, IEEE, 2011.

[88] Z. Deng, Q. Gu, and Q. Li, "Perceptually consistent example-based human motion retrieval," in *Symposium on Interactive 3D Graphics SI3D*, pp. 191–198, ACM, 2009.

[89] F. Patrona, A. Chatzitofis, D. Zarpalas, and P. Daras, "Motion analysis: Action detection, recognition and evaluation based on motion capture data," *Pattern Recognition*, vol. 76, pp. 612–622, 2018.

[90] H. Wu, J. Shao, X. Xu, Y. Ji, F. Shen, and H. T. Shen, "Recognition and detection of two-person interactive actions using automatically selected skeleton features," *IEEE Trans. Hum. Mach. Syst.*, vol. 48, no. 3, pp. 304–310, 2018.

[91] M. Meshry, M. E. Hussein, and M. Torki, "Linear-time online action detection from 3d skeletal data using bags of gesturelets," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9, IEEE Computer Society, 2016.

[92] S. Y. Boulahia, É. Anquetil, F. Multon, and R. Kulpa, "Cudi3d: Curvilinear displacement based approach for online 3d action detection," *Comput. Vis. Image Underst.*, vol. 174, pp. 57–69, 2018.

[93] K. Papadopoulos, E. Ghorbel, R. Baptista, D. Aouada, and B. E. Ottersten, "Two-stage rgb-based action detection using augmented 3d

poses," in *18th Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*, pp. 26–35, Springer, 2019.

[94] M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, and A. D. Bimbo, "Motion segment decomposition of RGB-D sequences for human behavior understanding," *Pattern Recognition*, vol. 61, pp. 222–233, 2017.

[95] A. Sharaf, M.Torki, M. E. Hussein, and M. El-Saban, "Real-time multi-scale action detection from 3d skeleton data," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 998–1005, IEEE Computer Society, 2015.

[96] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2752–2759, IEEE Computer Society, 2013.

[97] R. Liu, J. Shen, H. Wang, C. Chen, S. ching Cheung, and V. K. Asari, "Enhanced 3d human pose estimation from videos by using attention-based neural network with dilated convolutions," *International Journal of Computer Vision*, 2021.

[98] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5693–5703, IEEE, 2019.

[99] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11977–11986, IEEE, 2019.

[100] T. Alldieck, M. A. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll, "Learning to reconstruct people in clothing from a single RGB camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019.

[101] A. Aristidou, D. Cohen-Or, J. K. Hodgins, and A. Shamir, "Self-similarity analysis for motion capture cleaning," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 297–309, 2018.

[102] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "Rgb-d-based human motion recognition with deep learning: A survey," *Computer Vision and Image Understanding*, vol. 171, pp. 118–139, 2018.

[103] E. H. Jacox and H. Samet, "Metric space similarity joins," *ACM Transactions on Database Systems*, vol. 33, no. 2, 2008.

[104] B. Morschheuser, J. Hamari, J. Koivisto, and A. Maedche, "Gamified crowdsourcing: Conceptualization, literature review, and future agenda," *International Journal of Human-Computer Studies*, vol. 106, pp. 26–43, 2017.

[105] J. Li, X. Xie, Q. Pan, Y. Cao, Z. Zhao, and G. Shi, "Sgm-net: Skeleton-guided multimodal network for action recognition," *Pattern Recognition*, pp. 1–38, 2020.

[106] R. Baeza-Yates and B. A. Ribeiro-Neto, *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[107] V. Mic, D. Novak, and P. Zezula, "Binary sketches for secondary filtering," *ACM Transactions on Information Systems*, vol. 37, no. 1, 2018.

[108] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search: The Metric Space Approach*, vol. 32 of *Advances in Database Systems*. Springer-Verlag, 2006.

Part II

Collection of Works

Work 1

Sedmidubsky, J., Elias, P., and Zezula, P. Effective and Efficient Similarity Searching in Motion Capture Data. *Multimedia Tools and Applications*, 77(10), pp. 12073–12094, Springer, 2018.

Contribution of Jan Sedmidubský

≈ 50 % contribution: participation in proposal of main paper idea (transformation of 3D actions into 2D motion images and extraction of corresponding deep features); development of the deep-feature extractor; evaluation of experiments; participation in writing the text part of the paper

Work 2

Sedmidubsky, J., Budikova, P., Dohnal, V., and Zezula, P. Motion Words: A Text-like Representation of 3D Skeleton Sequences. In *42nd European Conference on Information Retrieval (ECIR)*, pp. 527–541, Springer, 2020.

Contribution of Jan Sedmidubský

$\approx$ 40 % contribution: participation in proposal of main paper idea (text-like representation of complex motion data); development of evaluation scenarios; participation in evaluation of experiments; participation in writing the text part of the paper

Work 3

Budikova, P., Sedmidubsky, J., Horvath, J., and Zezula, P. Towards Scalable Retrieval of Human Motion Episodes. In *22nd International Symposium on Multimedia (ISM)*, pp. 49–56, IEEE Computer Society, 2020.

Contribution of Jan Sedmidubský

$\approx$ 40 % contribution: participation in proposal of main paper idea (matching of medium-sized motion episodes); development of deep-feature extractor for motion episodes; participation in evaluation of experiments; participation in writing the text part of the paper

Work 4

Sedmidubsky, J. and Zezula, P. Probabilistic Classification of Skeleton Sequences. In *29th International Conference on Database and Expert Systems Applications (DEXA)*, pp. 50–65, Springer, 2018.

Contribution of Jan Sedmidubský

$\approx$ 95 % contribution: proposal of main paper idea (confusion-based classifier); development of the proposed algorithm; evaluation of experiments; writing the text part of the paper

Work 5

Sedmidubsky, J. and Zezula, P. Efficient Combination of Classifiers for 3D Action Recognition. *Multimedia Systems*, 12 p., Springer, 2021.

Contribution of Jan Sedmidubský

$\approx$ 95 % contribution: proposal of main paper idea (efficient evaluation of fusion of classifiers); development of the proposed approach; evaluation of experiments; writing the text part of the paper

Work 6

Sedmidubsky, J., Valcik, J., and Zezula, P. A Key-Pose Similarity Algorithm for Motion Data Retrieval. In *15th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pp. 669–681, Springer, 2013.

Contribution of Jan Sedmidubský

$\approx$ 60 % contribution: proposal of main paper idea (efficient matching of query-relevant subsequences within long motions); development of the proposed algorithm; evaluation of experiments; participation in writing the text part of the paper

Work 7

Sedmidubsky, J., Zezula, P., and Svec, J. Fast Subsequence Matching in Motion Capture Data. In *21st European Conference on Advances in Databases and Information Systems (ADBIS)*, pp. 50–72, Springer, 2017.

Contribution of Jan Sedmidubský

$\approx$ 85 % contribution: proposal of main paper idea (subsequence search algorithm); development of the proposed algorithm; evaluation of experiments; participation in writing the text part of the paper

Work 8

Sedmidubsky, J., Elias, P., and Zezula, P. Searching for Variable-Speed Motions in Long Sequences of Motion Capture Data. *Information Systems*, 80, pp. 148–158, Elsevier, 2019.

Contribution of Jan Sedmidubský

$\approx$ 65 % contribution: proposal of main paper idea (efficient subsequence search algorithm capable of finding slower and faster query-relevant motions); participation in development of the proposed algorithm; evaluation of experiments; participation in writing the text part of the paper

Work 9

Elias, P., Sedmidubsky, J., and Zezula, P. A Real-Time Annotation of Motion Data Streams. In *19th IEEE International Symposium on Multimedia (ISM)*, pp. 154–161, IEEE Computer Society, 2017.

Contribution of Jan Sedmidubský

≈ 35 % contribution: participation in proposal of main paper idea (segment-based annotation of long motions); participation in writing the text part of the paper

Work 10

Carrara, F., Elias, P., Sedmidubsky, J., and Zezula, P. LSTM-Based Real-Time Action Detection and Prediction in Human Motion Streams. *Multimedia Tools and Applications*, 78(19), pp. 27309–27331, Springer, 2019.

Contribution of Jan Sedmidubský

$\approx$ 30 % contribution: participation in proposal of main paper idea (frame-based annotation of continuous human motion streams); participation in writing the text part of the paper