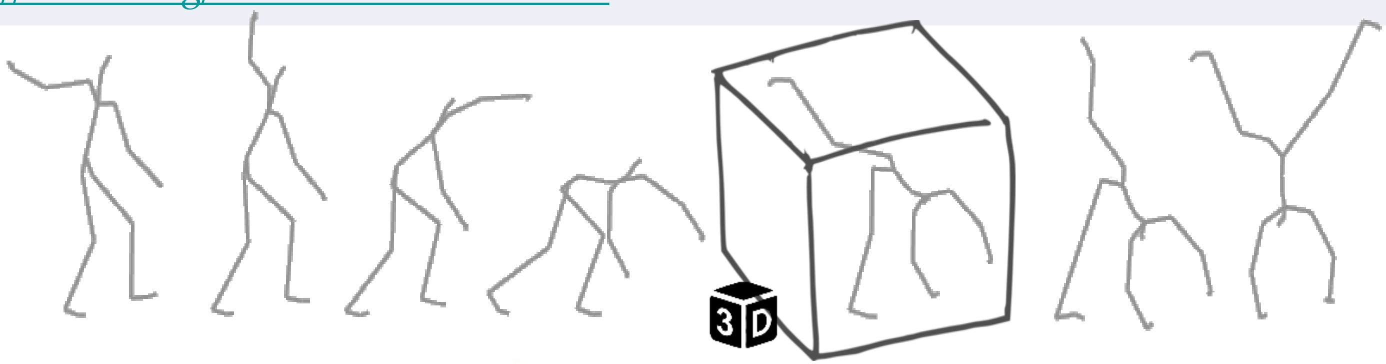
ACM MM 2018 Korea

Similarity-Based Processing of Motion Capture Data

Jan Sedmidubsky
xsedmid@fi.muni.cz

Pavel Zezula
zezula@fi.muni.cz

[Jan Sedmidubsky and Pavel Zezula. Similarity-Based Processing of Motion Capture Data. ACM Multimedia (MM 2018). ACM, pp. 2087–2089, 2018.]
https://dl.acm.org/citation.cfm?id=3241468



Laboratory of Data Intensive Systems and Applications
disa.fi.muni.cz

Faculty of Informatics, Masaryk University, Czech Republic
fi.muni.cz

Supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

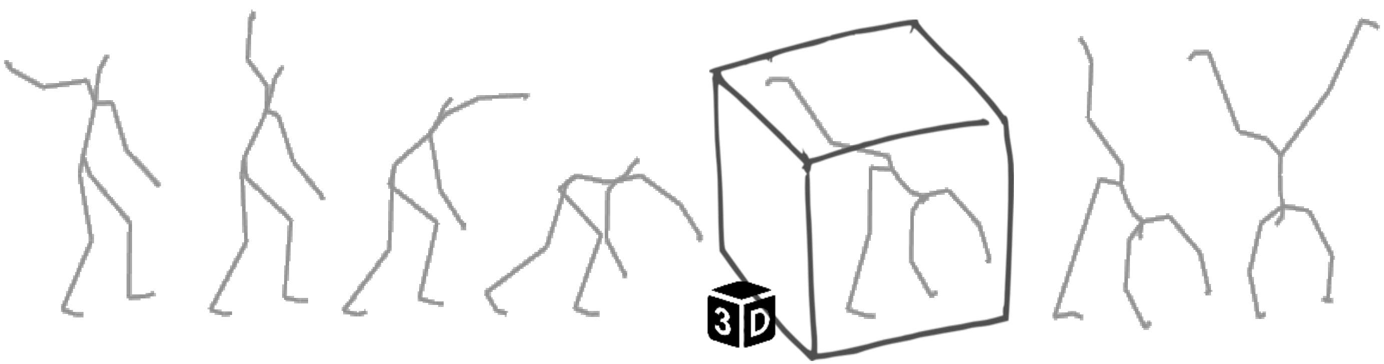
Outline

1) Motion Data: Acquisition and Applications
2) Challenges in Computerized Motion Data Processing
3) Similarity as a General Concept of Data Understanding
4) Similarity of Motion Sequences

- Coffee break -

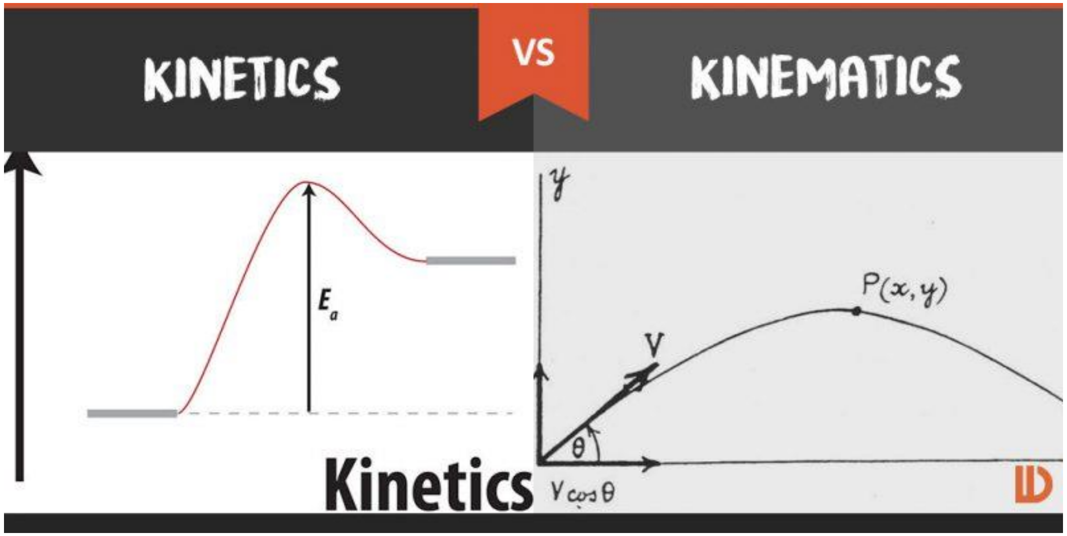
5) Classification of Segmented Motions
6) Processing Long and Unsegmented Motion Sequences
   - Subsequence Searching in Long Sequences
   - Stream-based Event Detection
7) Conclusions and Discussion

1 Motion Capture Data: Acquisition and Applications



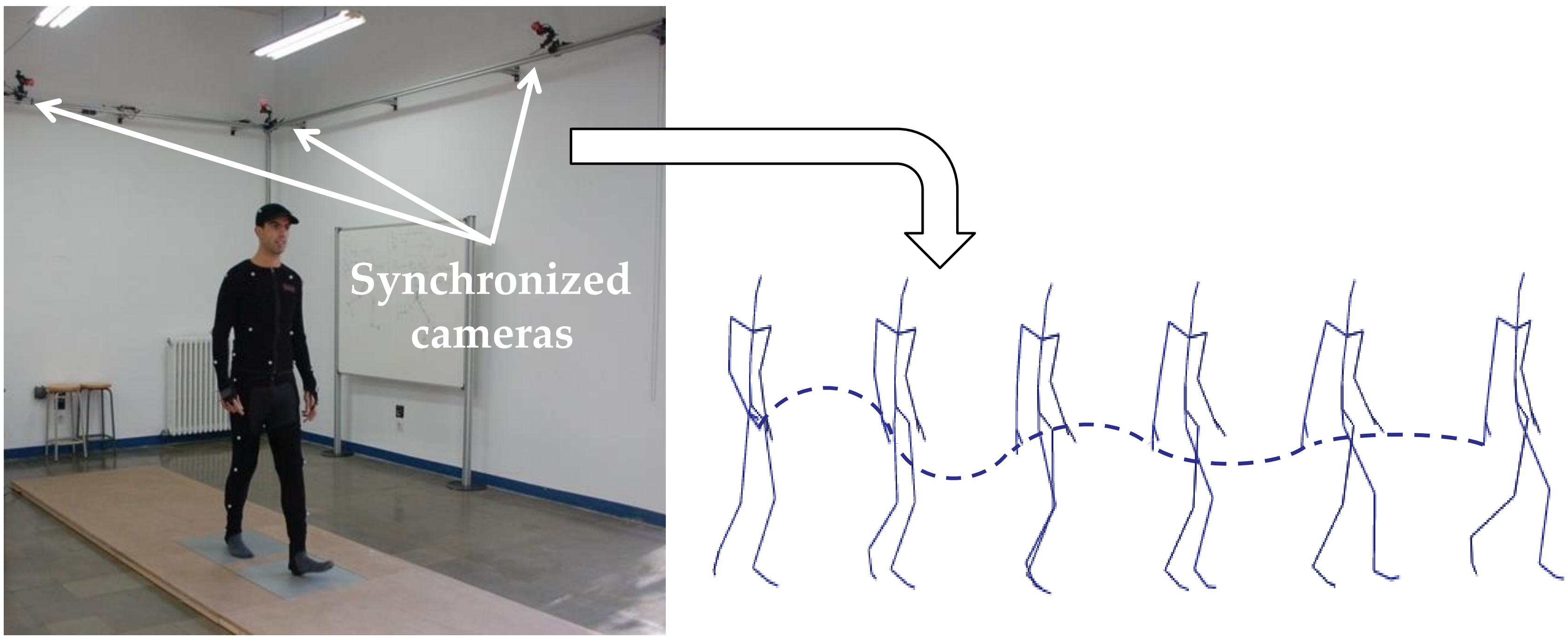
Motion data

- A digital representation of a human motion
- Types of data:
  - **Kinematic** – motion capture data, recorded by synchronized cams
  - **Kinetic** – ground-reaction force data, obtained by pressure plates



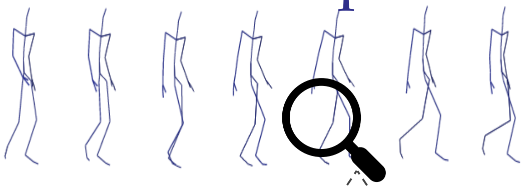
Motion Capture Data ~ MoCap Data ~ Motion Data

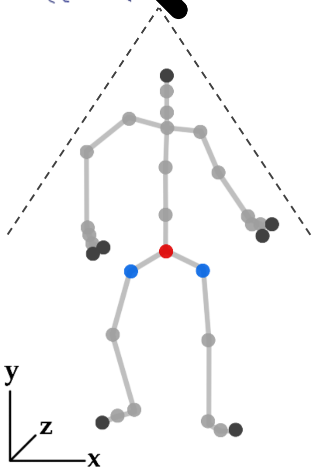
- Spatio-temporal 3D representation of a human motion



Motion capture data

- Continuous spatio-temporal characteristics of a human motion simplified into a discrete sequence of skeleton poses
  - Skeleton pose:
    - Skeleton configuration at a given time moment
    - 3D positions of body landmarks, denoted as joints





Pose captured in a given time moment

- Different views on motion data:
  - A sequence of skeleton poses
  - A set of 3D trajectories of joints

1.2 Capturing Devices

ACM MM 2018 Korea

Types of capturing devices

- Optical
  - Marker-based (invasive)
  - Marker-less (non-invasive)
- Inertial
- Magnetic
- Mechanical
- Radio frequency

1.2 Capturing Devices

ACM MM 2018 Korea

Accuracy of capturing devices



Device	Range [m]	Framerate [Hz]	Invasive	View field [°]	Tracked subjects	Positional accuracy [mm]	Rotational accuracy [°]	Landmark count
Kinect v1	0.8—4	30	No	57	2	50—150	?	20
Kinect v2	0.5—4.5	30	No	70	6	?	1—3	25
ASUS Xtion	0.8—3.5	30	No	58	?	?	?	?
Vicon MX40	space 7x7	120	Markers	360	?	0.063	?	32
Xsens MVN	?	120	Sensors	?	1	-	0.5—1	22
Organic Motion	space 4.3x3.8	120	No	360	5	1	1—2	22

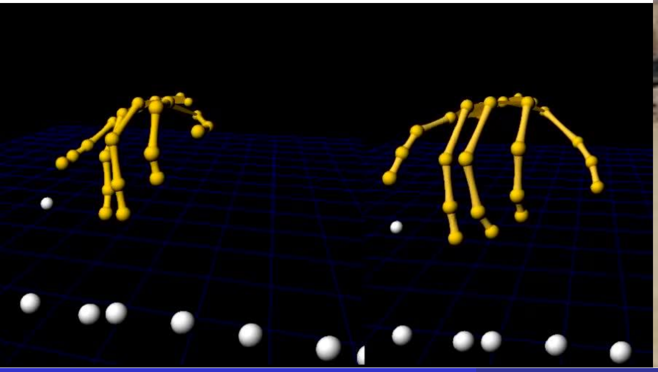
Capturing devices

- Optical-based devices are the most commonly used
- Advantages/disadvantages:
  - Invasive – accurate | large space | markers | expensive
    - Vicon, MotionAnalysis
  - Non-invasive – no markers | small space
    - Accurate but expensive – Organic Motion
    - Less accurate but cheap – Microsoft Kinect, ASUS Xtion
- Hardware devices and applicable software tools are usually independent
  - iPi Soft – marker-less, up to 16 cameras or 4 Kinects

- **Captured motion data serve as an input for our research**

Applications

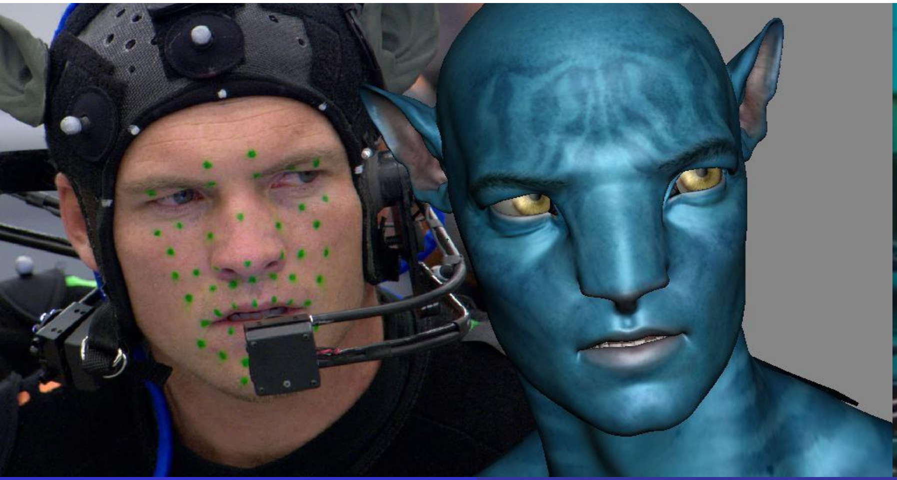
- Many application domains where motion data have a great potential to be utilized and automatically processed
  - Computer animation & human-computer interaction
  - Military
  - Sports
  - Medicine
  - Other domains

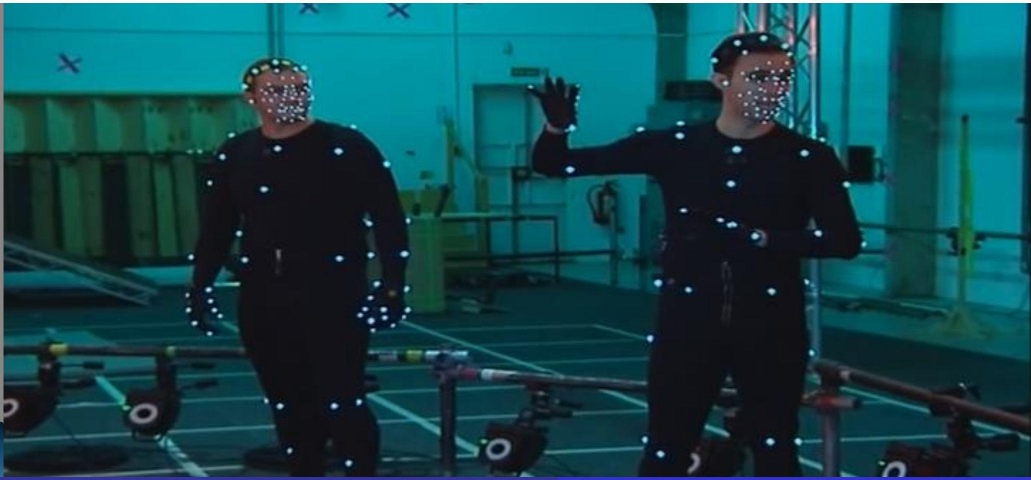


1.3 Applications

Computer animation

- Make subject (human) movements in movies and computer games as much realistic as possible
  - Games: Far Cry 4, GTA V
  - Movies: Avatar, The Lord of the Rings
- Create/generate new motions by merging movements that follow each other





Human computer interaction, augmented reality

- Detection of gestures/actions to enable real-time interactions

1.3 Applications

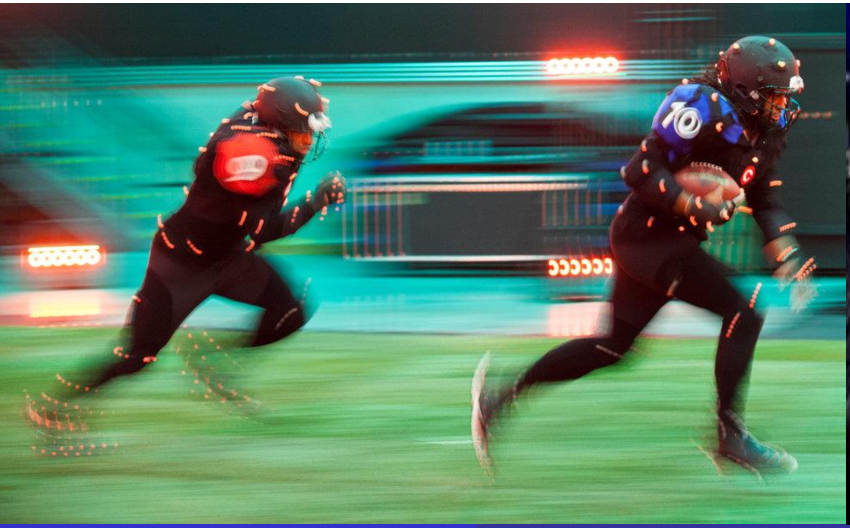
Military

- Interaction with digitally animated characters in live training scenarios in a natural and intuitive way
- Simulation of a combat and conflict-resolving situations
  - To improve the education and training of military forces or healthcare personnel by inserting live role-players

1.3 Applications

Sports

- Digital referees – detection of fouls
- Digital judges – assignment of scores
- Movement analysis to quantify an improvement or loss of performance

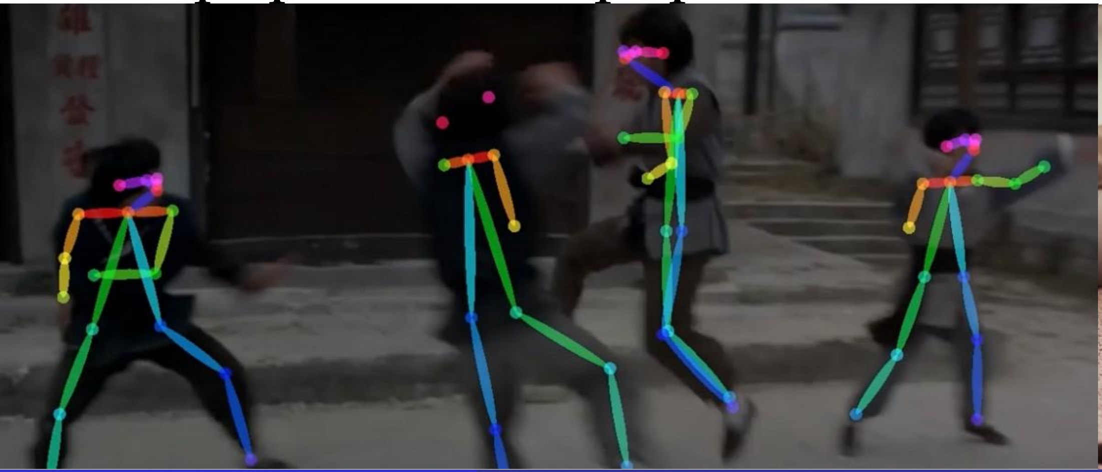


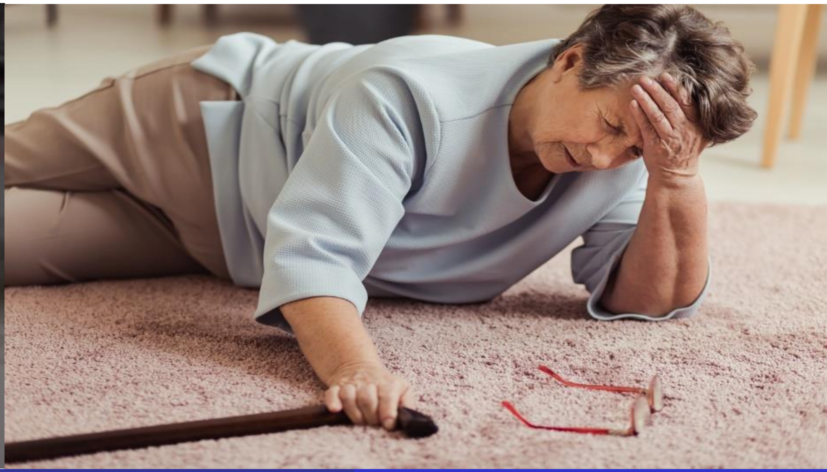
Medicine

- Improvement of the education and training of healthcare personnel including physicians, paramedics and nurses
- Creation of a roadmap to help each patient by showing exactly where and how he or she has gotten better
- Recognition of developmental disabilities or movement disorders

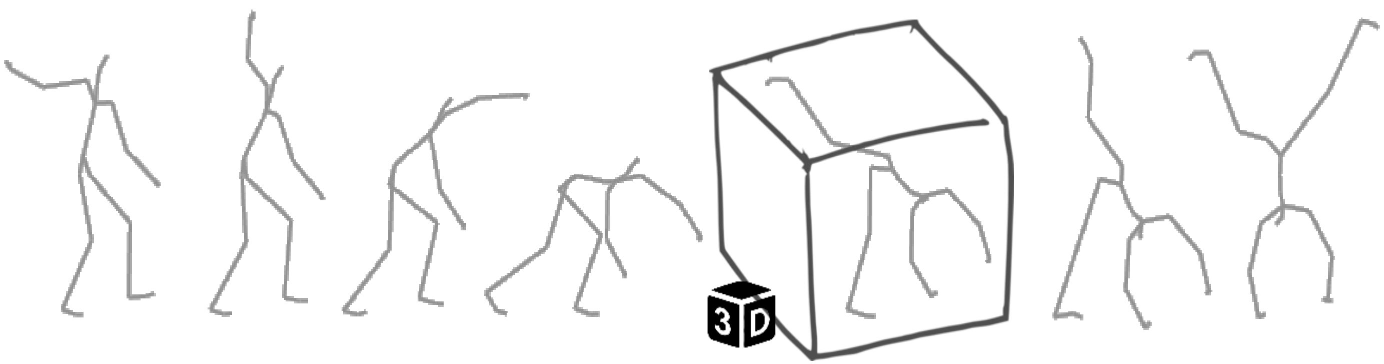
Other domains

- Law enforcement – identification of persons based on their style of walking
- Smart-homes – detection of falls of elderly people
- Construction-sites – identification of unsafe acts, e.g., speed limit violations of equipment or close proximity between equipment or equipment and workers





2 Challenges in Computer-Aided Processing



2 The Big Data Corollaries

Shifts in thinking

- From *some to all* – more scalability
- From *clean to messy* – less determinism (ranked comparisons)
- Loads on a sharp rise – usage on decline

Foundational concerns

- *Scalable* and *secure* data *analysis*, *organization*, *retrieval*, and *modeling*

Technological obstacles

- *Heterogeneity*, *scale*, *timeliness*, *complexity*, and *privacy* aspects

2 The Big Data Corollaries

ACM MM 2018 Korea

The (3V) problem: Volume, Variety, Velocity

- Issues:
  - Acquisition – what to keep and what to discard
  - Datafication – render into data aspects that do not exist in analog form
  - Unstructured data – structured only on storage and display
  - Inaccuracy – approximation, imprecision, noise

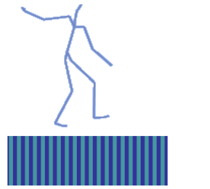
Motion data specifics

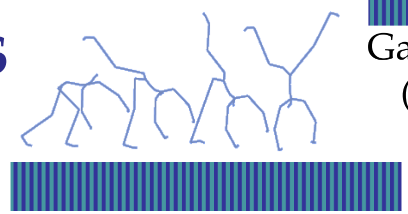
- Large volume of data
  - E.g., 31 joints · 3D space · 120 Hz => 11,160 float numbers/second generated => 1.5 TB/year needed to store the data
- Inaccuracy of data – captured data can be:
  - Inconsistent (e.g., location of markers)
  - Imprecise (e.g., inaccurate information about positions of joints)
  - Incomplete (e.g., missing information about some joint positions)
- Variety of motion-analysis operations
  - Designing operations, such as similarity comparison, searching, classification, semantic segmentation, clustering or outlier detection, with respect to the spatio-temporal nature of motion data

2.1 Data – Types of Motions

Motion data types

- **Short** motions:
  - Semantically-**indivisible** motions ~ **ACTIONS**
  - Length – typically in order of seconds
  - Database – usually a large number of actions
- **Long** motions:
  - Semantically-**divisible** motions ~ sequences of actions
  - Length – in order of minutes, hours, days, or even unlimited
  - Database – typically a single long motion processed either as a whole, or in the stream-based nature


Gait cycle (0.6 s)


Cartwheel (2.1 s)

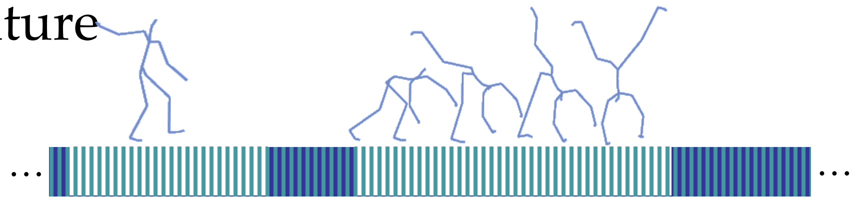
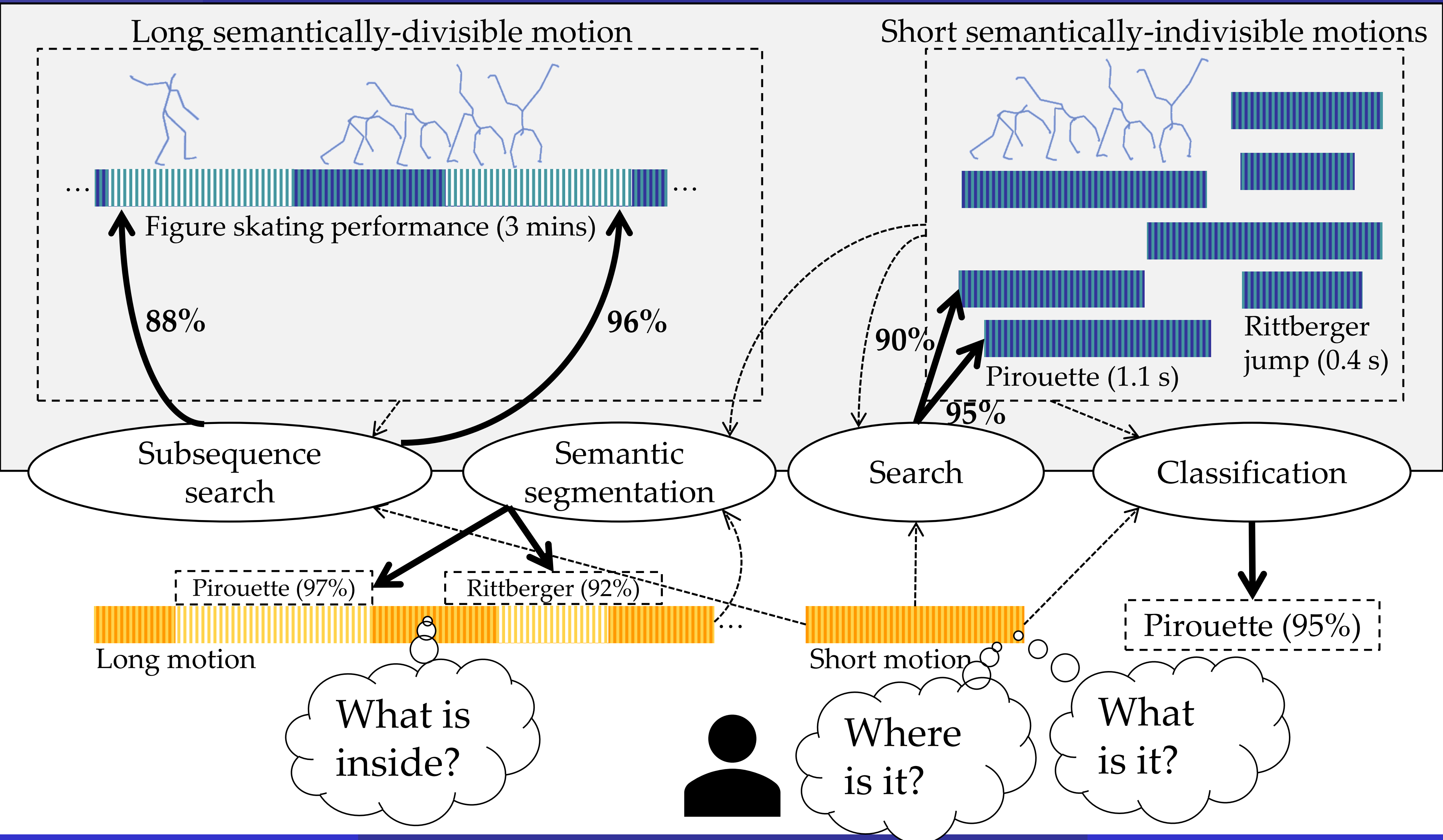


Figure skating performance (3 mins)

2.3 Motion-Analysis Operations

ACM MM 2018 Korea



Motion-analysis operations

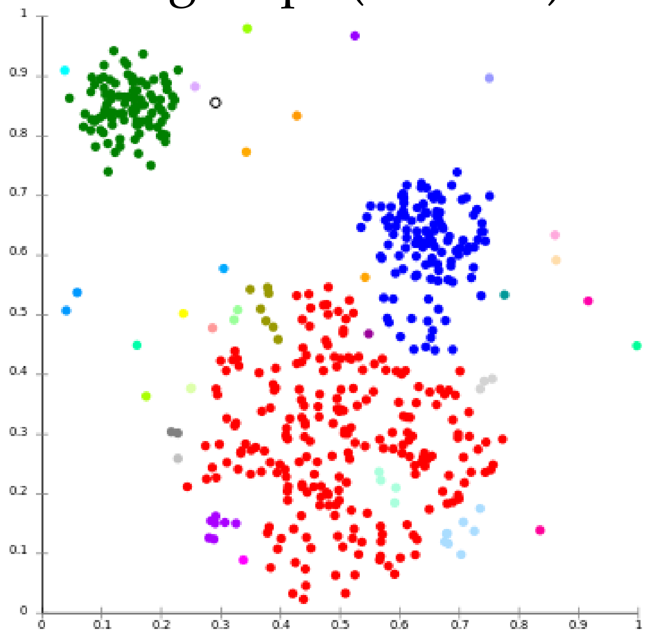
- Search
- Subsequence search
- Classification
- Semantic segmentation
- Other operations:
  - Clustering
  - Outlier detection
  - Joins
  - Mining frequent movement patterns
  - Action prediction

  ⋮

2.3 Other Operations – Clustering

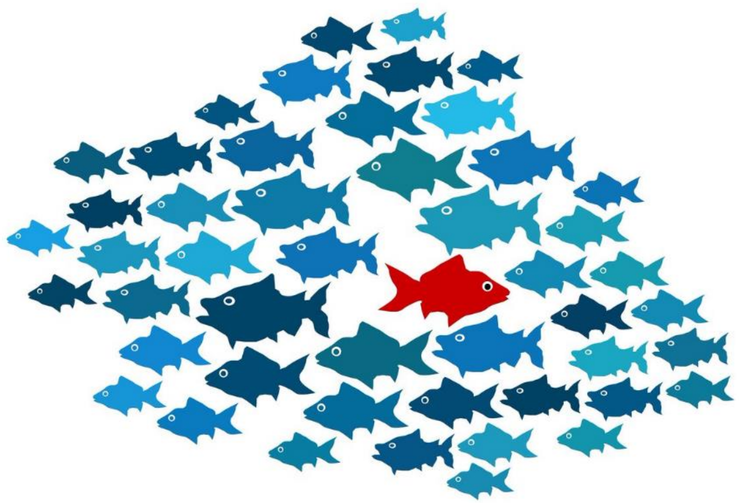
Clustering

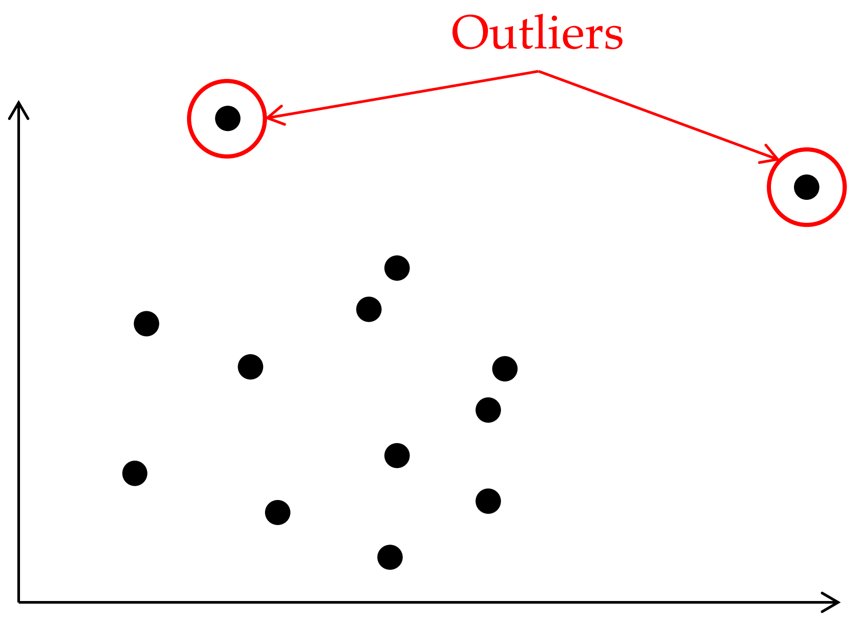
- Suppose each motion as a point in *n*-dimensional space
- Grouping motions in action collections
  - Motions in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)
- Useful for statistical data analysis



Outlier detection

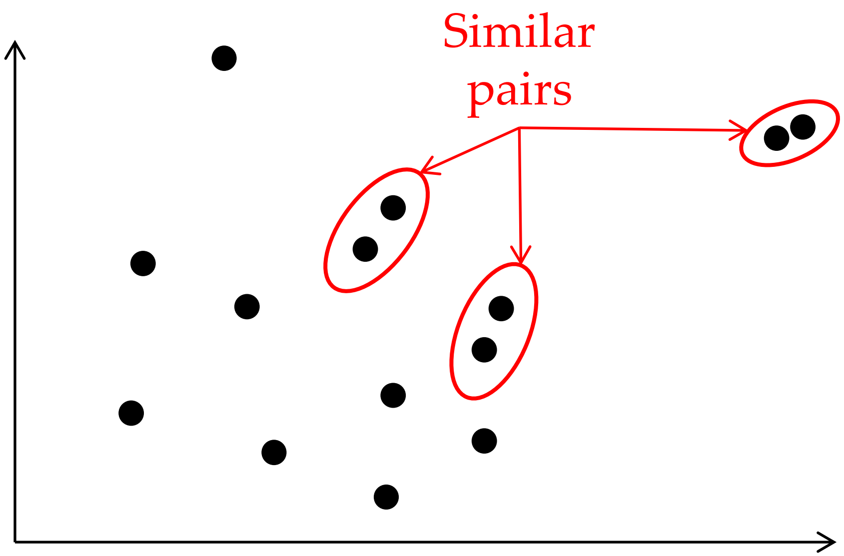
- Identifying motions which significantly deviate from other motion entities





Similarity join

- Finding pairs of similar motions
- Types:
  - Range joins – finding all the motion pairs at distance at most r
  - k-closest pair joins – finding the k closest motion pairs



2.3 Summary of Motion-Analysis Operations

ACM MM 2018 Korea

Summary of operations

OPERATION	OPERATION DATA (KNOWLEDGE BASE)	USER INPUT	OPERATION RESULT
Search	Unannotated actions	Query action	Actions similar to the query action
Subsequence search	Unannotated long motions	Query action	Beginnings/endings of query-similar subsequences
Classification	Labelled (categorized) actions	Action	Class of examined action
Semantic segmentation	Labelled (categorized) actions	Long motion	Beginnings/endings of detected and recognized actions

Require annotated (labeled) data

=> All the operations require the concept of motion similarity

3 Similarity as a General Concept of Data Understanding

3.1 Real-Life Motivation

The social psychology view

- Any event in the history of organism is, in a sense, unique
- *Recognition*, *learning*, and *judgment* presuppose an ability to categorize stimuli and classify situations by similarity
- Similarity (*proximity*, *resemblance*, *communality*, *representativeness*, *psychological distance*, etc.) is fundamental to theories of *perception*, *learning*, *judgment*, etc.
- Similarity is subjective a context-dependent

Are they similar?

Are they similar?

Are they similar?

Are they similar?

3.1 Contemporary Networked Media

The digital data point of view

- Almost everything that we *see*, *read*, *hear*, *write*, *measure*, or *observe* can be digital
- Users autonomously *contribute* to production of global media and the growth is exponential
- Sites like Flickr, YouTube, Facebook host user contributed content for a variety of events
- The elements of networked media are related by numerous multi-facet links of similarity

Challenge

- Networked media database is getting close to the human "fact-bases"
  - The gap between physical and digital world has blurred

- Similarity data management is needed to *connect*, *search*, *filter*, *merge*, *relate*, *rank*, *cluster*, *classify*, *identify*, or *categorize* objects across various collections

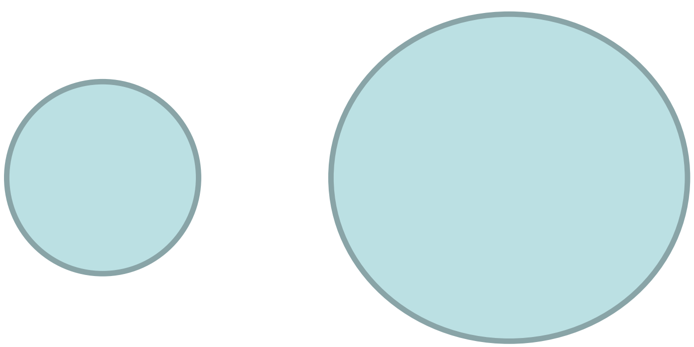
WHY?

It is the *similarity* which is in the world *revealing*

Similarity in geometry

- Figures that have the same shape but not necessarily the same size are similar figures
- Any two line segments are similar:

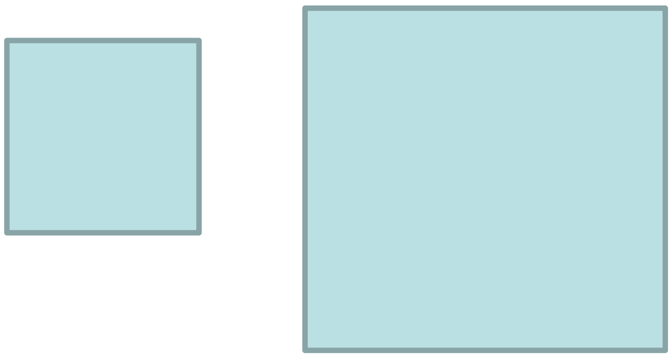
- Any two circles are similar:



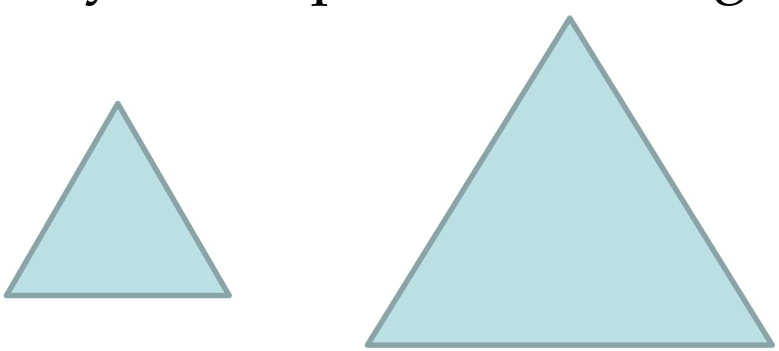
3.1 Similarity in Geometry

Similarity in geometry

- Any two squares are similar:

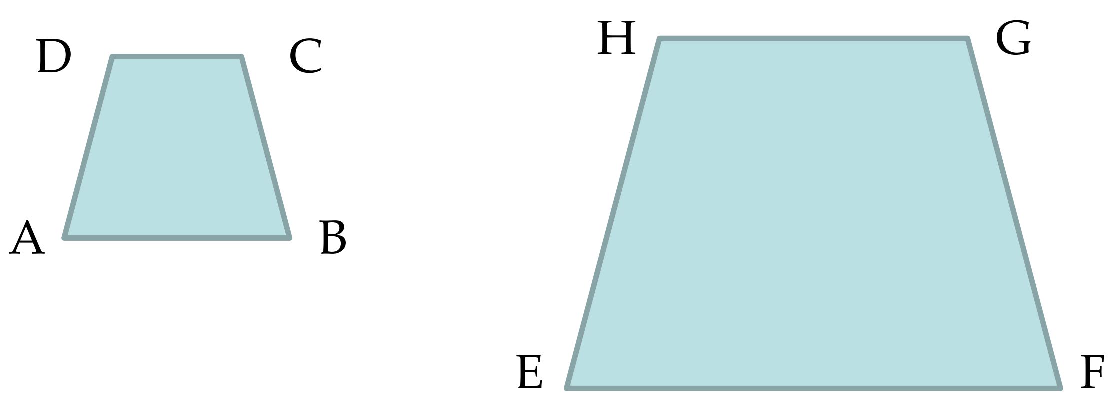


- Any two equilateral triangles are similar:



Similarity in geometry

- Two polygons are similar to each other, if:
  1) Their corresponding angles are congruent
     - ∠A = ∠E; ∠B = ∠F; ∠C = ∠G; ∠D = ∠H, and
  2) The lengths of their corresponding sides are proportional
     - AB/EF = BC/FG = CD/GH = DA/HE



Similarity in geometry

- If one polygon is similar to a second polygon, and the second polygon is similar to the third polygon, the first polygon is similar to the third polygon

- In any case: two geometric figures are either similar, or they are not similar at all

3.2 Metric Space Model of Similarity

Metric space $\mathcal{M} = (\mathcal{D}, d)$

- $\mathcal{D}$ – domain of objects
- $d(x, y)$ – distance function between objects x and y
  - $\forall\, x, y, z \in \mathcal{D}$:

$$d(x, y) > 0 \qquad \text{– non-negativity}$$
$$d(x, y) = 0 \Leftrightarrow x = y \qquad \text{– identity}$$
$$d(x, y) = d(y, x) \qquad \text{– symmetry}$$
$$d(x, y) \leq d(x, z) + d(z, y) \qquad \text{– triangle inequality}$$

Example of distance functions

- L_p Minkovski distance – for vectors
  - L_1 – city-block distance
  - L_2 – Euclidean distance
  - L_∞ – infinity

$$L_1(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

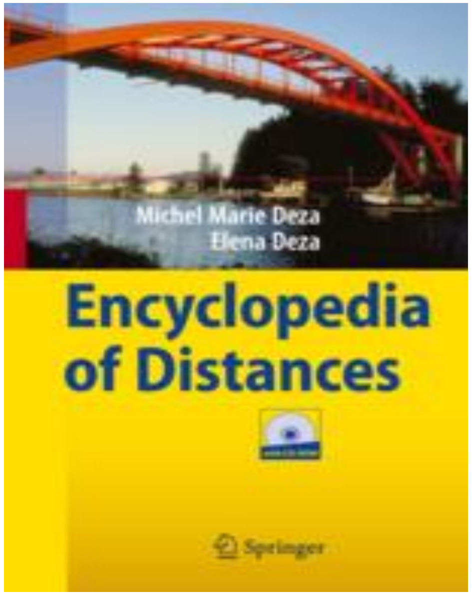
$$L_\infty(x,y) = \max_{i=1}^{n} |x_i - y_i|$$

- Edit distance – for strings
  - Minimum number of insertions, deletions and substitutions
  - d(“application”, “applet”) = 6
- Jaccard’s coefficient – for sets A, B

$$d(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Example of other distance functions

- Hausdorff distance
  - For sets with elements related by another distance
- Earth-movers distance
  - Primarily for histograms (sets of weighted features)
- Mahalanobis distance
  - For vectors with correlated dimensions
- and many others – see the book



3.2 Metric Space – Search Problem

Similarity search problem in metric spaces

- For $X \subseteq \mathcal{D}$ in metric space $\mathcal{M}$, pre-process X so that the similarity queries are executed efficiently

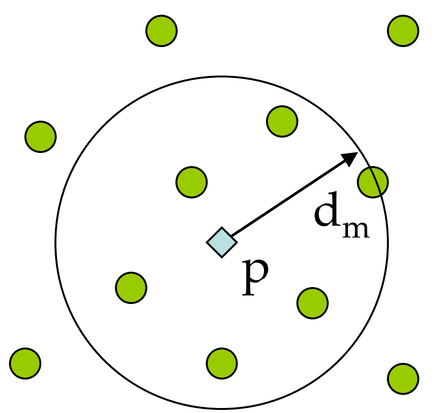
- In metric spaces:
  - No total ordering exists!
  - Queries only expressed by examples!

Basic partitioning principles

- For $X \subseteq \mathcal{D}$ in metric space $\mathcal{M} = (\mathcal{D}, d)$

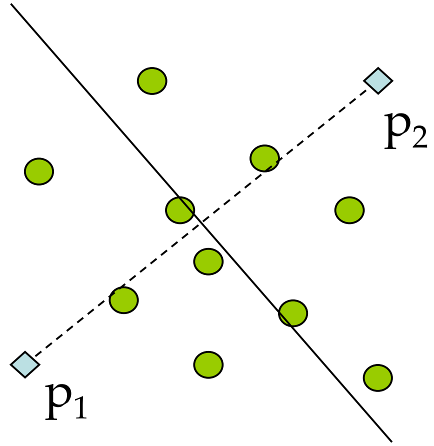
Ball partitioning

Inner set: $\{ x \in X \mid d(p, x) \leq d_m \}$
Outer set: $\{ x \in X \mid d(p, x) > d_m \}$



Generalized hyper-plane partitioning

$\{ x \in X \mid d(p_1, x) \leq d(p_2, x) \}$
$\{ x \in X \mid d(p_1, x) > d(p_2, x) \}$

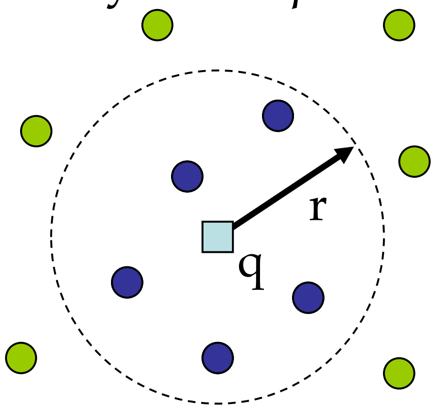


3.2 Metric Space – Similarity Queries

Range query

$R(q, r) = \{x \in X \mid d(q, x) \leq r\}$

"all museums up to 2km from my hotel *q*"



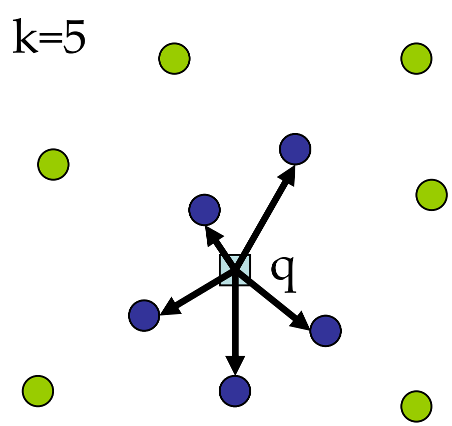
Nearest neighbor query

$NN(q) = \{x \in X \mid \forall y \in X, d(q,x) \leq d(q,y)\}$

k-nearest neighbor query

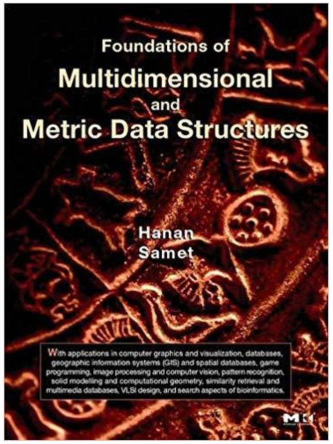
$k\text{-}NN(q, k) = A$
$A \subseteq X, |A| = k$
$\forall x \in A, y \in X - A, d(q, x) \leq d(q, y)$

"five closest museums to my hotel *q*"

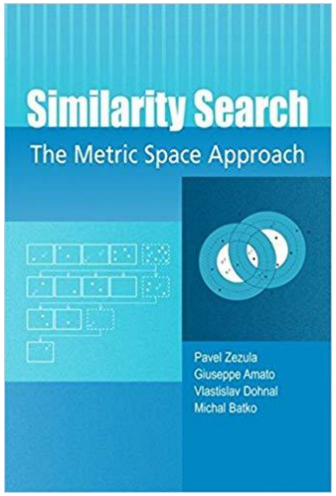


3.2 Similarity Search Textbooks

Major textbooks on metric searching technologies



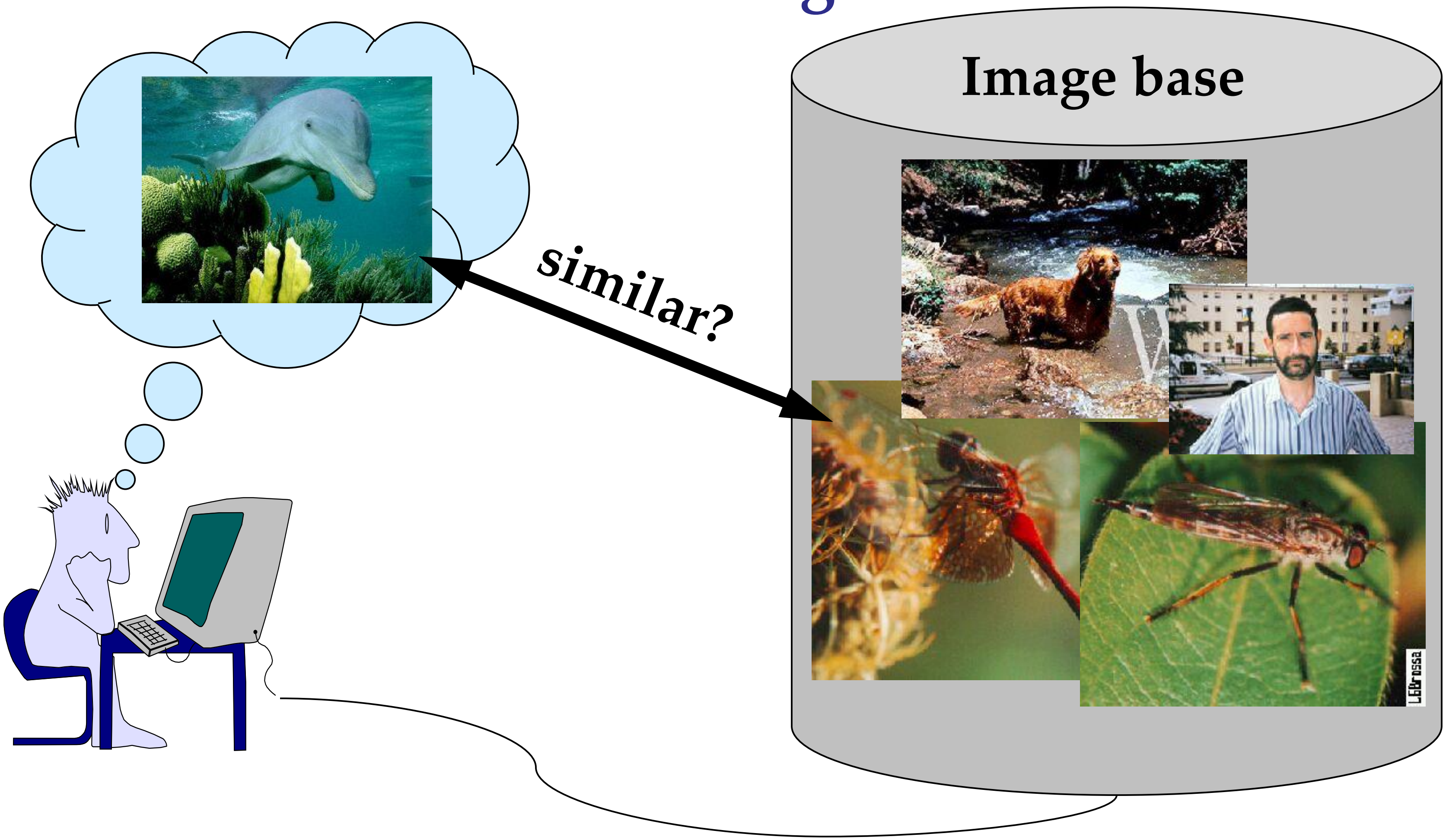
H. Samet
Foundation of Multidimensional and Metric Data Structures
Morgan Kaufmann, 1,024 pages, 2006



P. Zezula, G. Amato, V. Dohnal, and M. Batko
Similarity Search: The Metric Space Approach
Springer, 220 pages, 2005

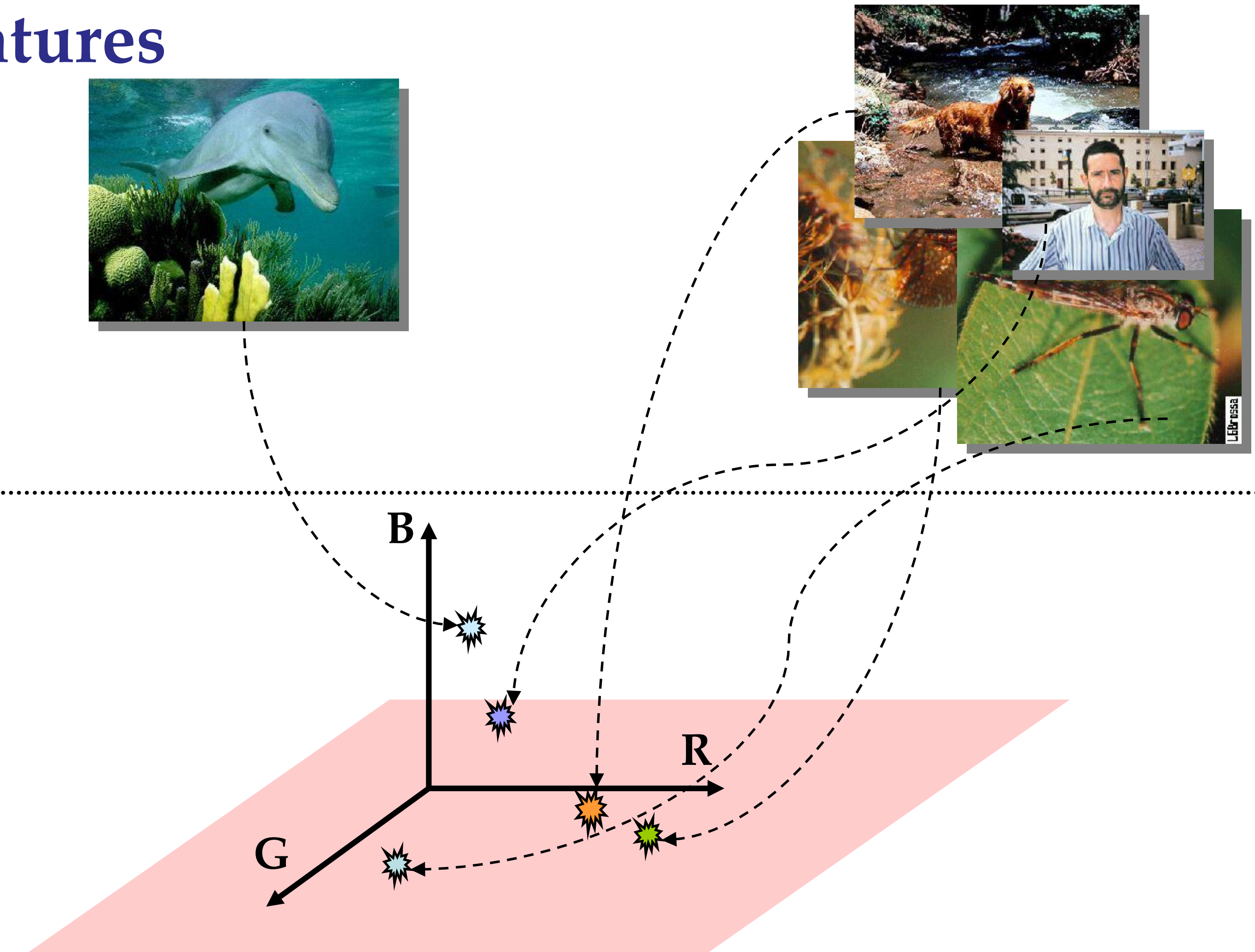
Teaching materials:
http://www.nmis.isti.cnr.it/amato/similarity-search-book/

Content-based search in images



Extracting features

Image level

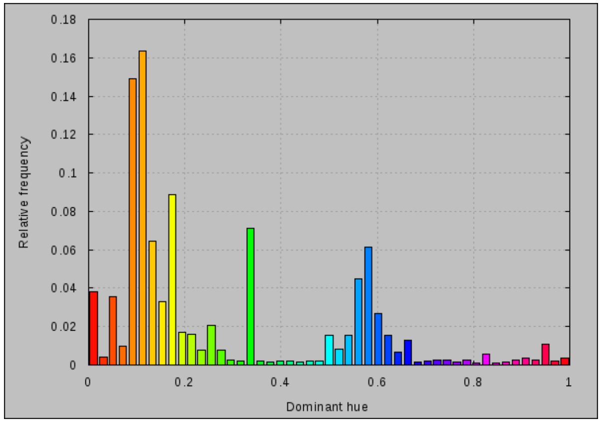


Feature level

3.2 Visual Similarity

Examples of features

- MPEG-7 multimedia content descriptor standard
  - Global feature descriptors – color, shape, texture, etc.
  - One high-dimensional (282 dimensions) vector per image



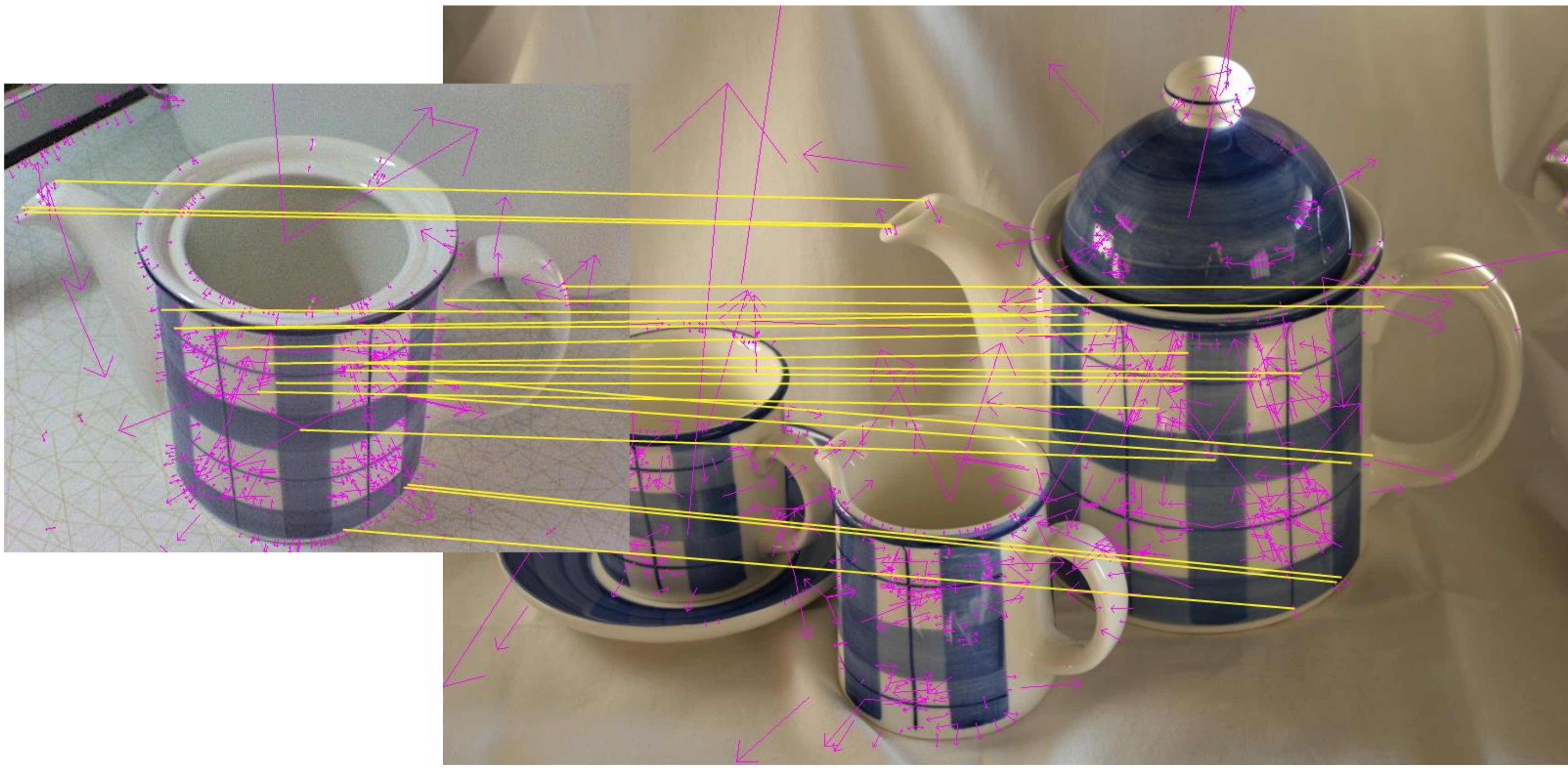
Multiple visual aspects

ACM MM 2018 Korea

Examples of features

- Local feature descriptors – SIFT, SURF, etc.
  - Invariant to image scaling, small viewpoint change, rotation, noise, illumination

Finding correspondence



Biometric similarity

- Biometrics – methods of recognizing a person based on physiological and/or behavioral characteristics
- Two types of recognition problems:
  - Verification – authenticity of a person
  - Identification – recognition of a person
- Examples:
  - Fingerprints, face, iris, retina, speech, gait, etc.

3.3 Applications – Biometrics

Fingerprints

- Minutiae detection:
  - Detect ridges (endings and branching)
  - Represented as a sequence of minutiae
    - P=((r_1, e_1, θ_1), …, (r_m, e_m, θ_m))
    - Point in polar coordinates (r, e) and direction θ
- Matching of two sequences:
  - Align input sequence with a database one
  - Compute a weighted edit distance
    - $w_{ins,\ del} = 620$
    - $w_{repl} = [0; 26]$ – depending on similarity of two minutiae

Hand recognition

- Hand image analysis
  - Contour extraction, global registration
    - Rotation, translation, normalization
  - Finger registration
  - Contour represented as a set of pixels $F = \{f_1, \ldots, f_{N_F}\}$
- Matching: modified Hausdorff distance

$$H(F,G) = \max(h(F,G), h(G,F))$$

$$h(F,G) = \frac{1}{N_F} \sum_{f \in F} \min_{g \in G} \|f - g\| \qquad h(G,F) = \frac{1}{N_G} \sum_{g \in G} \min_{f \in F} \|f - g\|$$

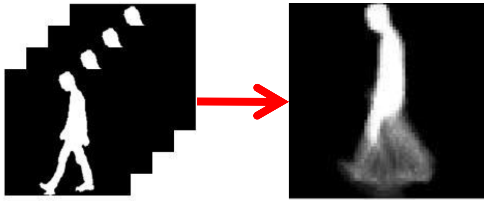
3.3 Applications – Remote Biometrics

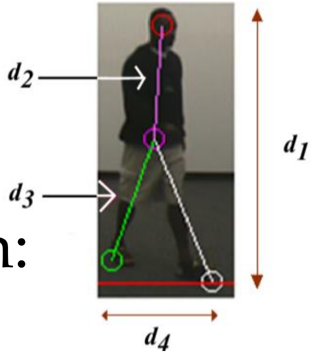
Recognition process

- Detection, normalization, extraction, recognition

Face recognition

- Methods:
  - Appearance-based – analyze the face as a whole
  - Model-based – compare individual features (e.g., eyes, mouth)





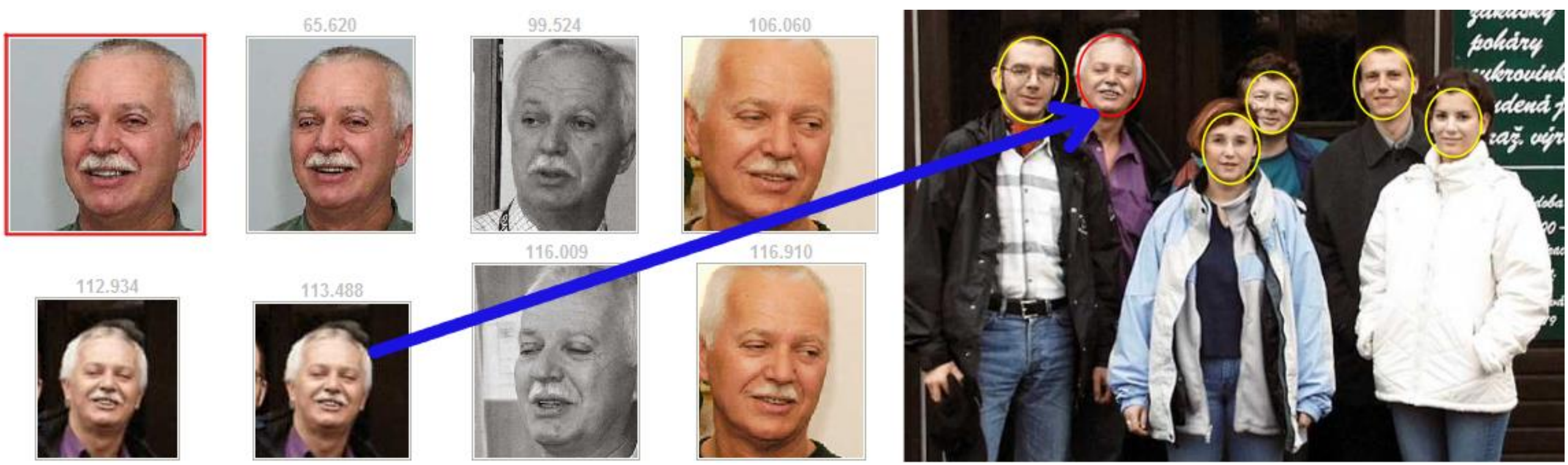
Gait recognition

- Methods based on shape or dynamics of the person:
  - Appearance-based – analyze person’s silhouettes
  - Model-based – compare features (e.g., trajectory, angular velocity)

3.3 Applications – Face Recognition

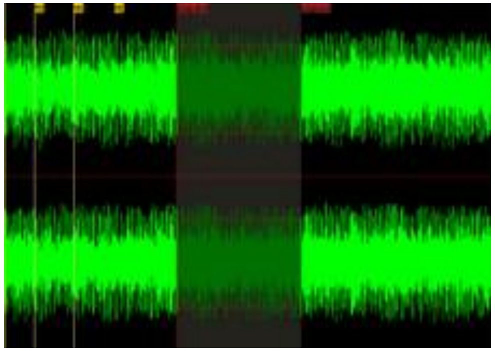
Face similarity

- Face detection
- Face recognition – distance function
- Similarity search in collections of face characteristics



Signal processing

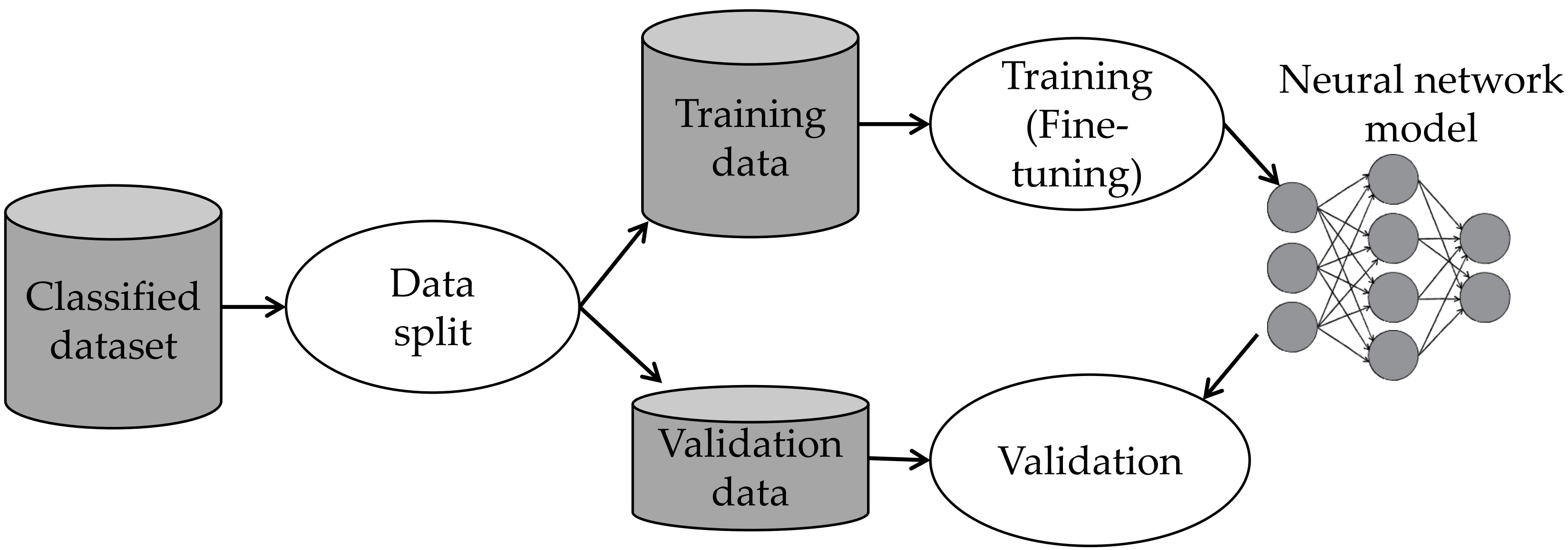
- Vast amount of signals produced:
  - Biomedicine data – ECG, CT, EEG, MR
  - Audio data – audio similarity, recognition
  - Financial time series – analysis, forecasting
  - Time series streams



- Demand for:
  - A graceful handling of such data
  - Flexible reactions to new application needs

Feature extraction

- Neural networks
  - Deep convolutional neural networks (DCNN)
  - Recurrent neural networks (RNN)



3.3 Applications – Demos

MUFIN similarity-search demos

- 20M images: http://disa.fi.muni.cz/demos/profiset-decaf/
- Fashion: http://disa.fi.muni.cz/twenga/
- Image annotation: http://disa.fi.muni.cz/annotation/

- Fingerprints: http://disa.fi.muni.cz/fingerprints/
- Time series: http://disa.fi.muni.cz/subseq/
- Multi-modal person ident.: http://disa.fi.muni.cz/mmpi/

3 SISAP Conference

SISAP (Similarity Search and Applications)

- International conference series (http://sisap.org/)

2009 Prague Czechia

2011 Lipari Italy

2013 A Coruña Spain

2015 Glasgow UK

2017 Munich Germany

2008 Cancun Mexico

2010 Instanbul Turkey

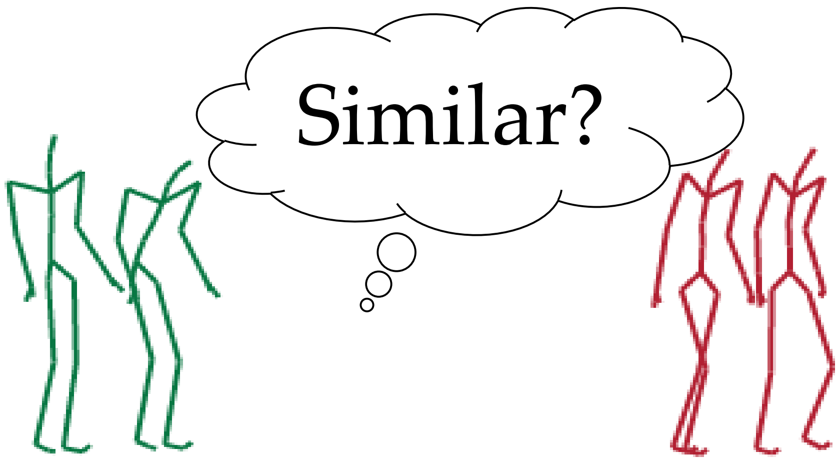
2012 Toronto Canada

2014 Los Cabos Mexico

2016 Tokyo Japan

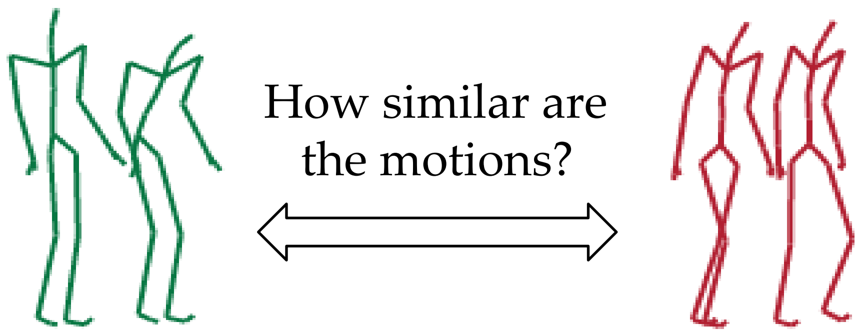
2018 Lima Peru

4 Similarity of Actions



Similarity of motions

- Determining similarity of motion sequences is an essential operation for computerized processing of motion data

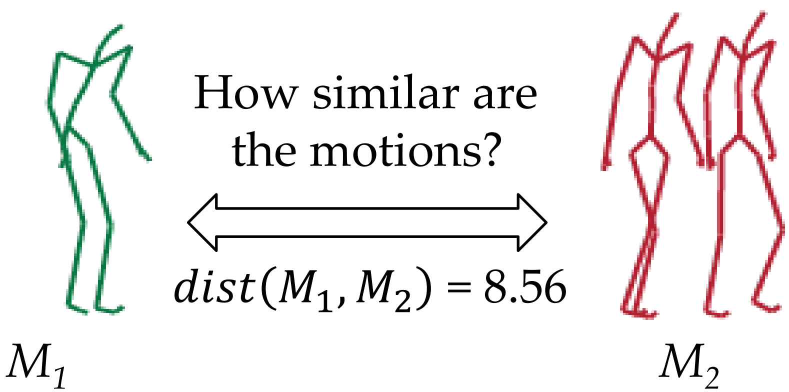


- Similarity is needed everywhere, e.g., for synthesis, clustering, searching, semantic segmentation

4.1 Similarity through Metric Spaces

Objective of similarity measures

- Develop an effective and efficient metric distance functions for quantifying similarity of actions
- Metric distance measure $dist(M_1, M_2) \rightarrow \boldsymbol{R}_0^+$
  - The value 0 means identical motions
  - The higher the value, the more dissimilar the motions are



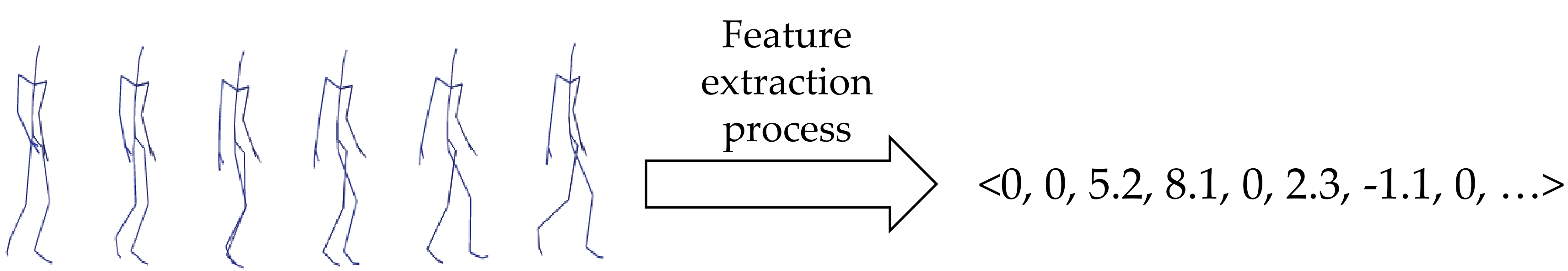
4.1 Challenges of Similarity Measures

Challenges

- Similarity is application-dependent (*e.g., recognizing daily actions vs. recognizing people based on their style of walking*)
- Subjects have different bodies *(e.g., child vs. adult)*
- The distance function needs to cope with spatial and temporal deformations
  - The same action (*e.g., kick*) can be performed at different:
    - Styles *(e.g., frontal kick vs. side kick)* and
    - Speeds *(e.g., faster vs. slower)*

Feature extraction and comparison

- Distance is very rarely evaluated on the captured skeleton sequences of 3D joint coordinates but rather on content-preserving features extracted from motions
  - A motion feature is usually represented as a set of time series or as a high-dimensional vector of real numbers
  - A motion feature is extracted in a pre-processing step



4.2 Types of Features

Granularity

- Pose-based features – a set of times series
- Motion-based features – a fixed-length vector

Space dependence

- Space-invariant features
- Space-dependent features

Engineering

- Hand-crafted features – manual feature engineering
- Machine-learned features – learning features automatically

Granularity of features

- Pose-based features – a set of times series
  - Each time series corresponds to specific characteristics computed for each pose (e.g., left-knee angle rotation)
  - Time-series length is equal to the number of poses (motion length)

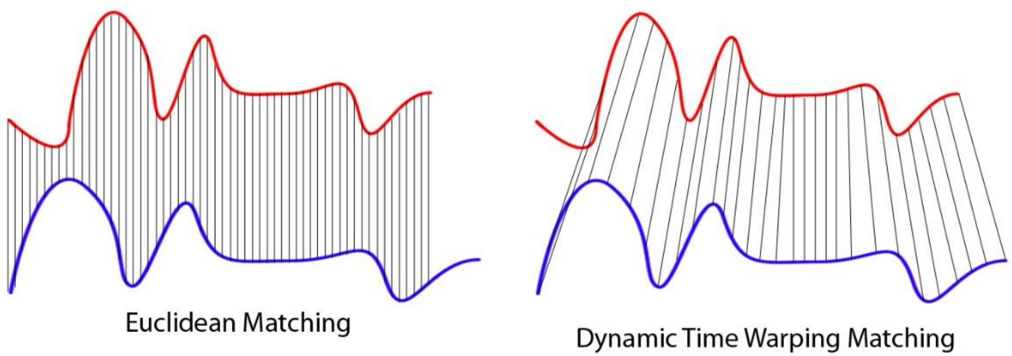
<4.2, 4.1, 4.0, 3.9, 3.8, 3.8, 3.7, 3.8, 3.9, 4.0, …>
<9.2, 9.1, 9.0, 9.9, 9.8, 9.8, 9.7, 9.8, 9.9, 9.0, …>
⋮

- Motion-based features – a fixed length vector
  - Vector dimensions correspond to aggregated/learned characteristics over the whole motion (e.g., average velocity of individual joints)

<0, 0, 5.2, 8.1, 0, 2.3, 1.1, 0.5>

Comparison of features

- Pose-based feat. – series of different lengths compared by:
  - Time-warping functions, e.g., Dynamic Time Warping (DTW)
  - Standard functions applied to normalized series in time dimension
    - Euclidean distance
    - Cosine distance



- Motion-based features – fixed-length vectors compared by standard functions:
  - Euclidean distance
  - Cosine distance

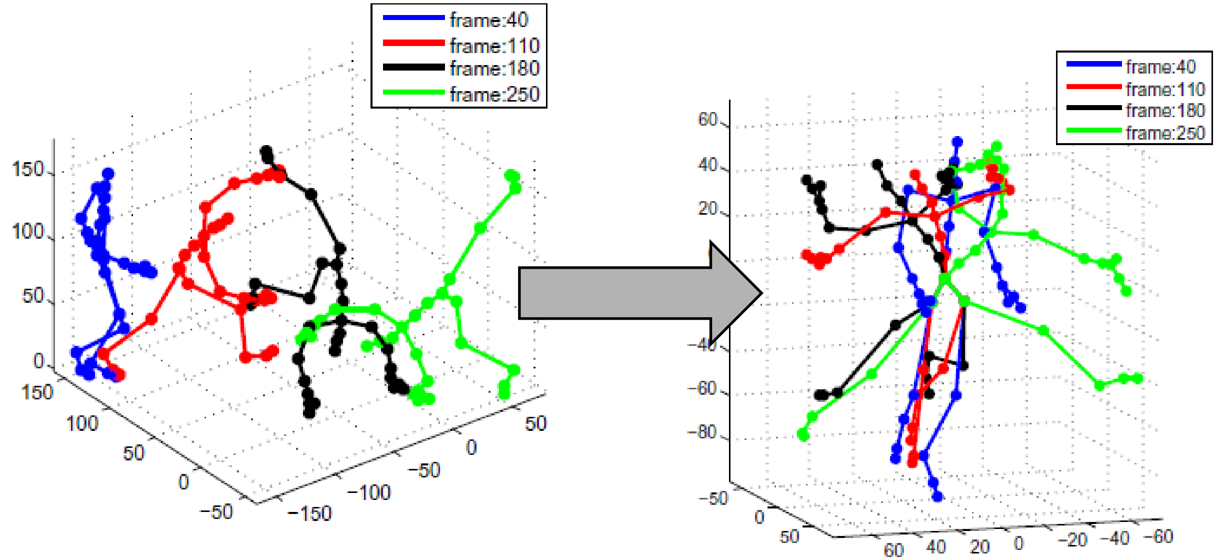
Feature dependence on a space

- Space-invariant features
  - Transformation from the original 3D space to a position-independent space
  - E.g., joint-angle rotations, distances between joints, velocities or accelerations of joints
- Space-dependent features
  - Feature values somehow related to the original 3D space
  - E.g., absolute or relative 3D joint positions
- Input data can be normalized before feature extraction

4.2 Input Data Normalization

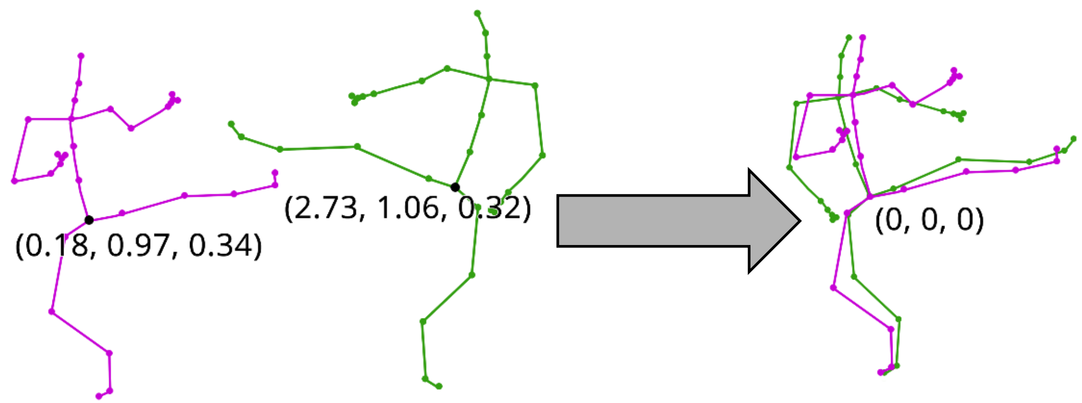
Normalization of:

- Position
- Orientation
- Skeleton size



Granularity:

- Single pose
- Whole motion



Feature engineering

- Developing a program (extractor) for extracting the features from input motions automatically
- Types of engineering:
  - Hand-crafted features
    - The program is manually developed by a domain expert
  - Machine-learned features
    - The program is automatically learned using a given machine-learning technique
    - Requires a large amount of categorized training data

"Coming up with features is difficult, time-consuming, requires expert knowledge." –Andrew Ng

4.2 Hand-Crafted Features

Hand-crafted features

- Very good knowledge of data domain is needed
- Very specialized in what they express

Existing hand-crafted-based approaches

- Classification of neurological disorders of gait
  - 17 scalars (e.g., gait velocity, stride length, step freq.)

    [Pradhan et al., Automated classification of neurological disorders of gait using spatio-temporal gait parameters, Journal of Electromyography and Kinesiology, 2015]
- Daily-activity search
  - 28 joint-angle rotations

    [Sedmidubsky et al., A key-pose similarity algorithm for motion data retrieval, 2013]
  - 40 relational frame-based characteristics

    [Muller et al., Efficient and robust annotation of motion capture data, 2009]

4.3 Learning Features

Feature learning

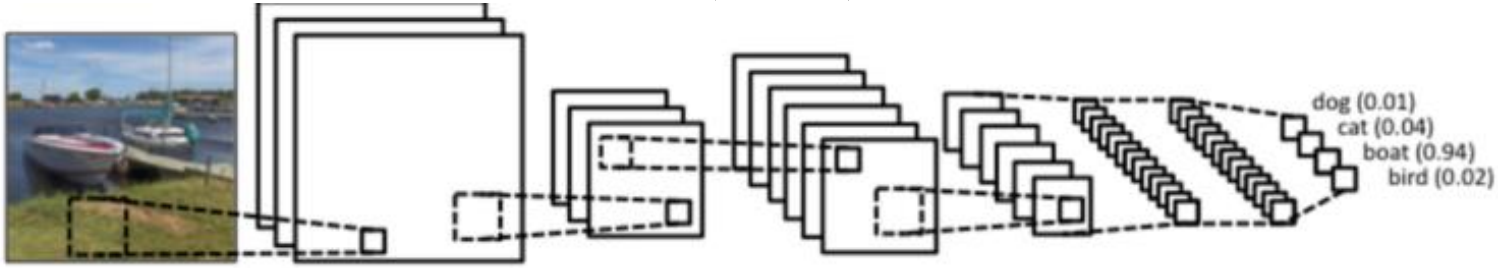
- Goal – utilizing machine-learning techniques to automatically discover the representations needed for feature detection or classification from input data
- Machine learning – a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed

Deep learning

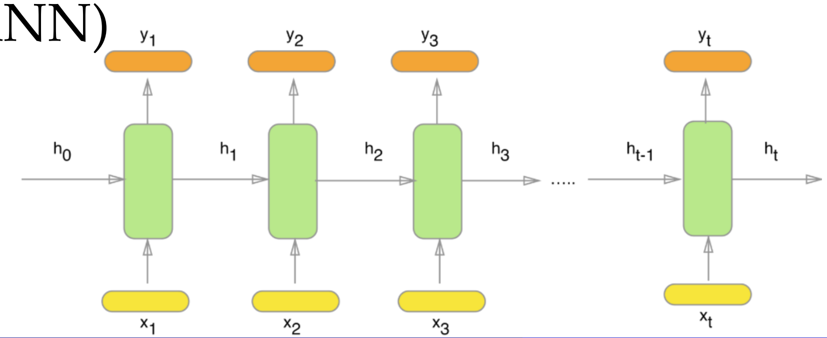
- Part of machine learning which derives meaning out of data by using a hierarchy of multiple layers that mimic the neural networks of our brain

Deep learning

- If large amounts of data are provided, the system begins to understand them and respond in useful ways
- Several architectures:
  - Convolutional neural networks (CNN)

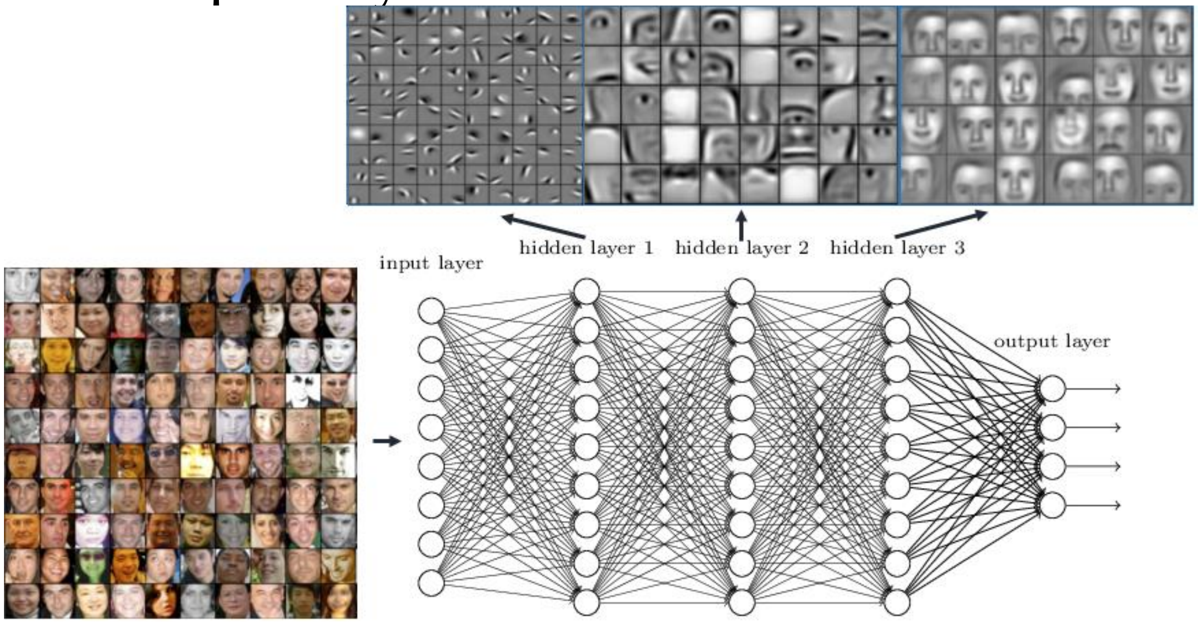


  - Recurrent neural networks (RNN)



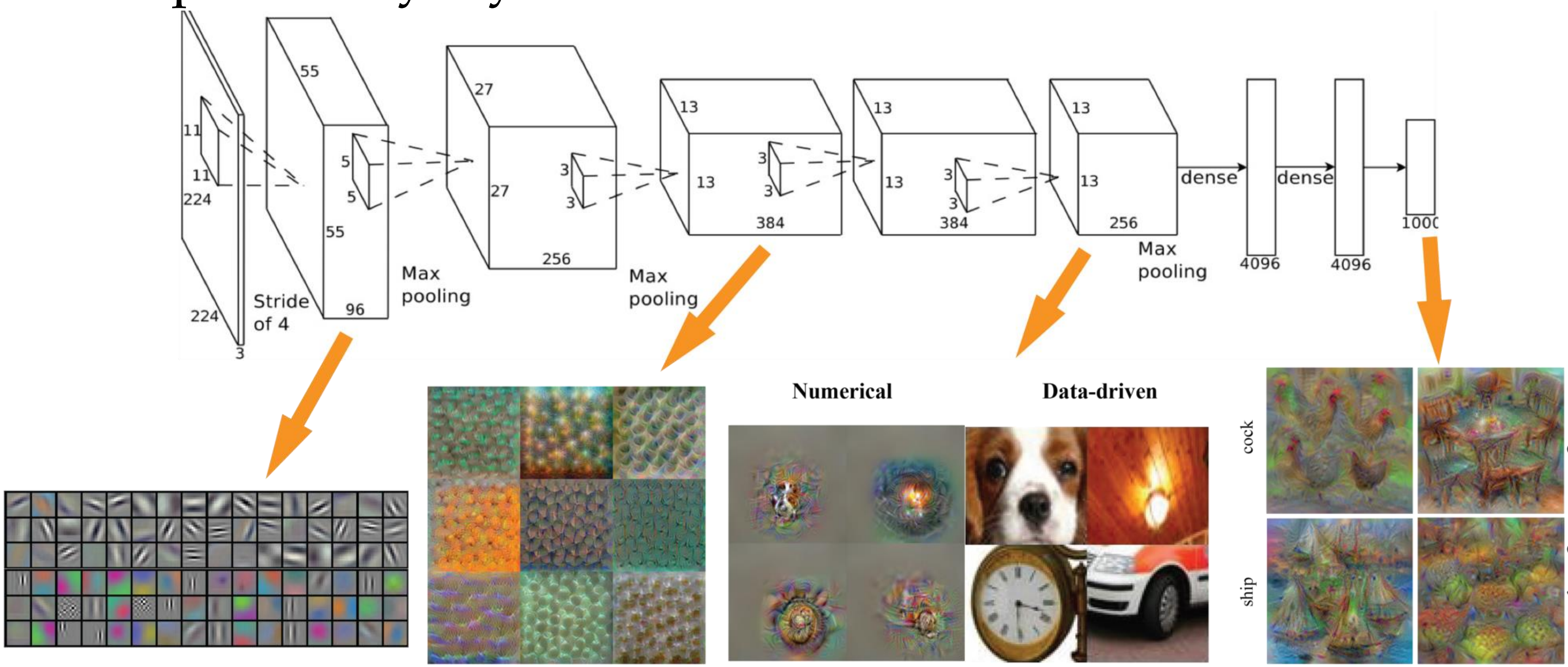
Convolutional neural networks (CNN)

- Consist of a hierarchy of layers
- Each layer transforms the data into more abstract representations (e.g., edge -> nose -> face)
- The output layer combines the features to make predictions



Convolutional neural network (CNN) – AlexNet

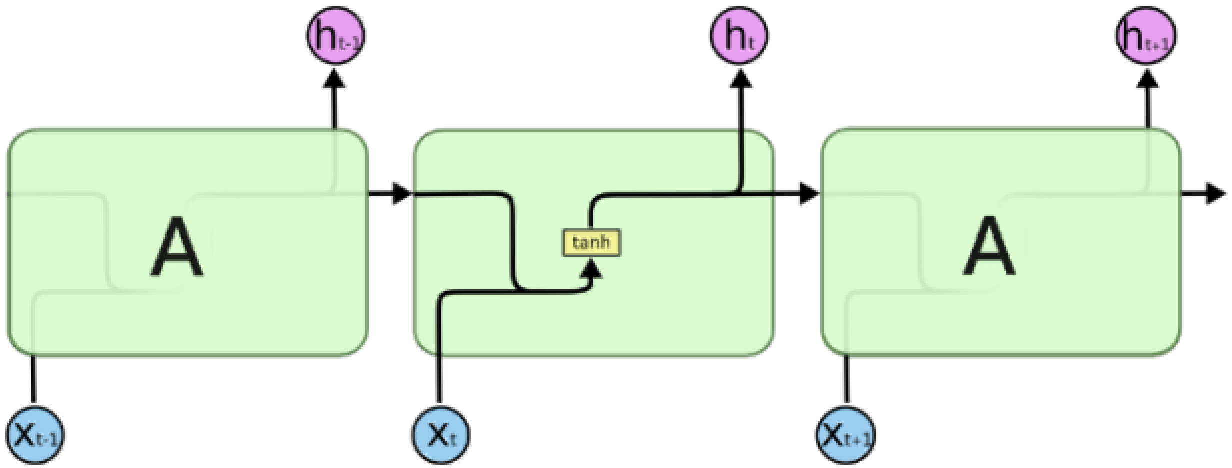
- The last layer with 1,000 output categories
- Output of any layer can be used as a feature



Conv 1: Edge+Blob | Conv 3: Texture | Conv 5: Object Parts | Fc8: Object Classes

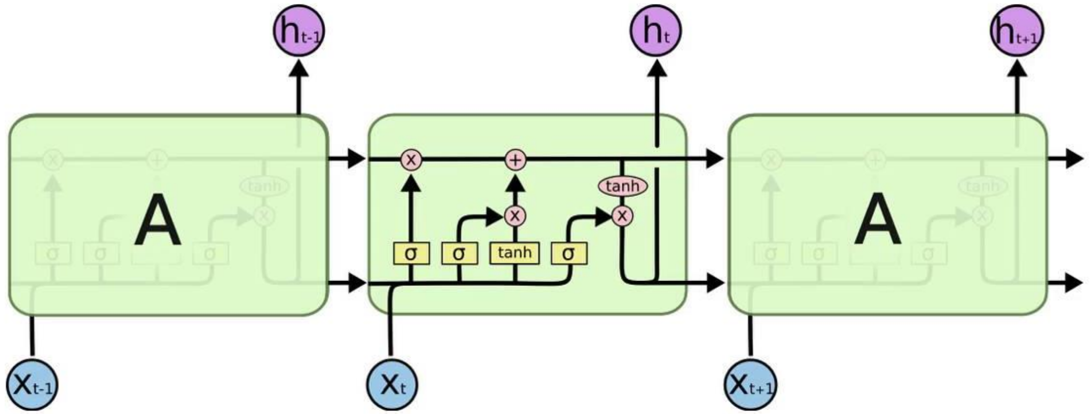
Recurrent neural networks (RNN)

- RNN cells remember the inputs in internal memory, which is very suitable for sequential data
- The output vector's contents are influenced by the entire history of inputs



Recurrent neural networks (RNN)

- Long-Short Term Memory (LSTM) networks:
  - Learn when data should be remembered and when they should be thrown away
  - Well-suited to learn from experience to classify, process and predict time series when there are very long time lags of unknown size between important events



Summary of deep learning

- It is no magic! Just statistics in a black box, but exceptional effective at learning patterns
- Excels in tasks where a basic unit (e.g., joint coordinate) has a very little meaning in itself, but the combination of such units has a useful meaning
- Requirements:
  - Measurable and describable goals (define the cost)
  - Large dataset of a good quality (input-output mappings)
  - Enough computing power (GPU instances)

Existing deep-learning approaches

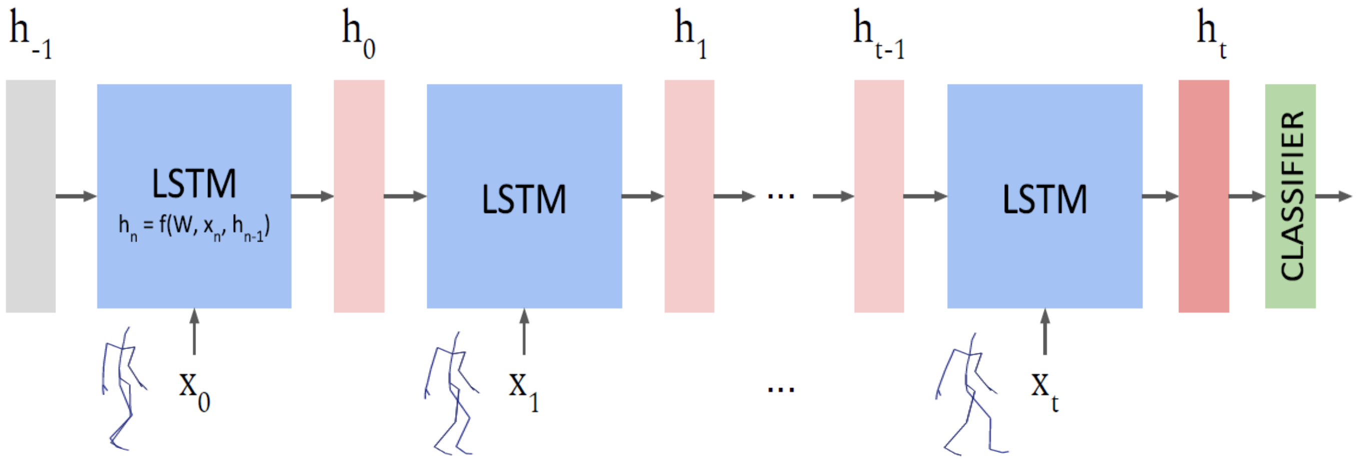
- Daily-activity classification
  - 16–256D float vectors compared by the Euclidean distance
    [Coskun et al.: Human Motion Analysis with Deep Metric Learning. ECCV, 2018]
  - 4,096D float vectors compared by the Euclidean distance
    [Sedmidubsky et al.: Probabilistic Classification of Skeleton Sequences. DEXA, 2018]
- Daily-activity search
  - 160D bit vectors compared by the Hamming distance
    [Wang et al.: Deep signatures for indexing and retrieval in large motion databases. Motion in Games, 2015]
  - 4,096D float vectors compared by the Euclidean distance
    [Sedmidubsky et al.: Effective and efficient similarity searching in motion capture data. Multimedia Tools and Applications, 2018]
- Person identification
  - 64D float vectors compared by the Euclidean distance
    [Coskun et al.: Human Motion Analysis with Deep Metric Learning. ECCV, 2018]

Advantages/disadvantages of features

	HAND-CRAFTED	MACHINE-LEARNED
Accuracy (descriptive power)		
Interpretability of dimensions		
Prerequisites	Very good scenario knowledge	Many example categorized motions
Application	More-easily describable scenarios	Most scenarios with some categorization

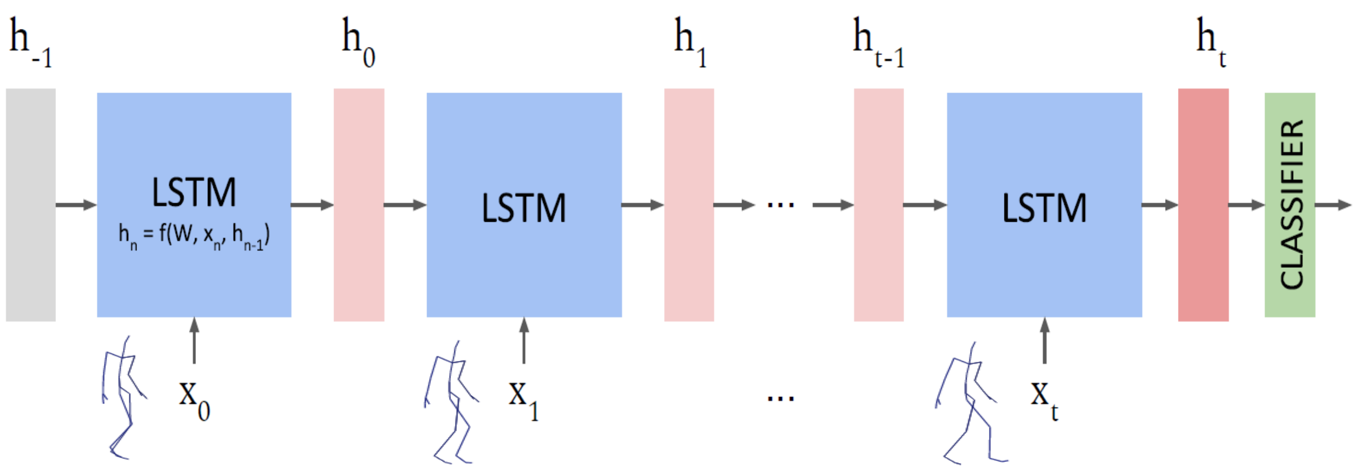
LSTM-based similarity concept

- Learning features based on classified training data
- LSTM network is ideal to model sequences of poses
- Sequence of LSTM cells, where output state depends on the current input and the previous state
  - Output state h_i of the i-th cell is fed to the next (i+1)-th cell
  - Number of states/cells corresponds to the number of poses (t)



LSTM-based similarity concept

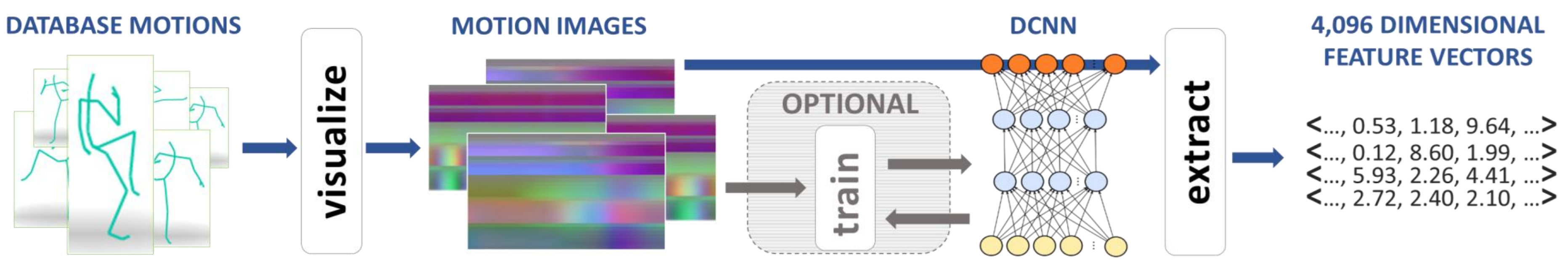
- The last state h_t can be used as a feature
- Size of each state h_i is a user-defined parameter
  - Suitable state size of 512 / 1,024 / 2,048 dimensions



Motion-image similarity concept

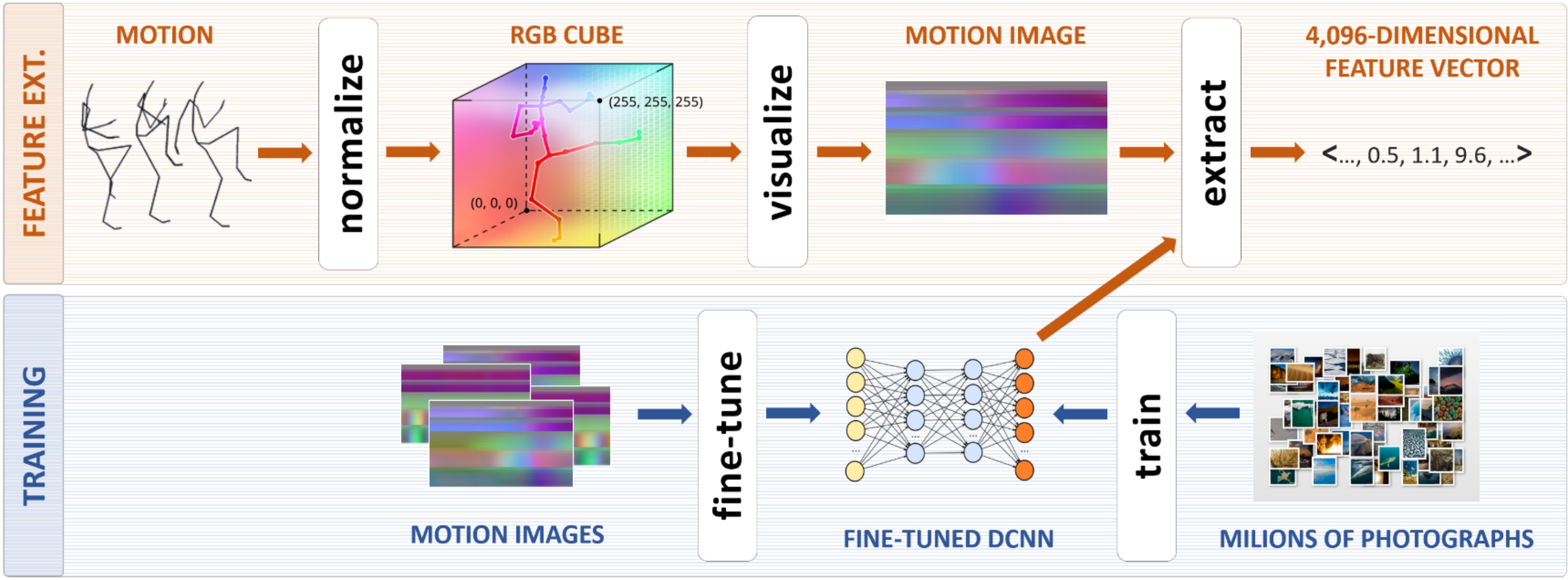
[Sedmidubsky et al.: Effective and efficient similarity searching in motion capture data. Multimedia Tools and Applications, 2018]

- Deep-learned 4,096D features compared by the Euclidean distance function
  - Very successfully evaluated in classification of daily activities
- Suitable for motions in order of seconds (e.g., gait cycles)



Feature extraction steps

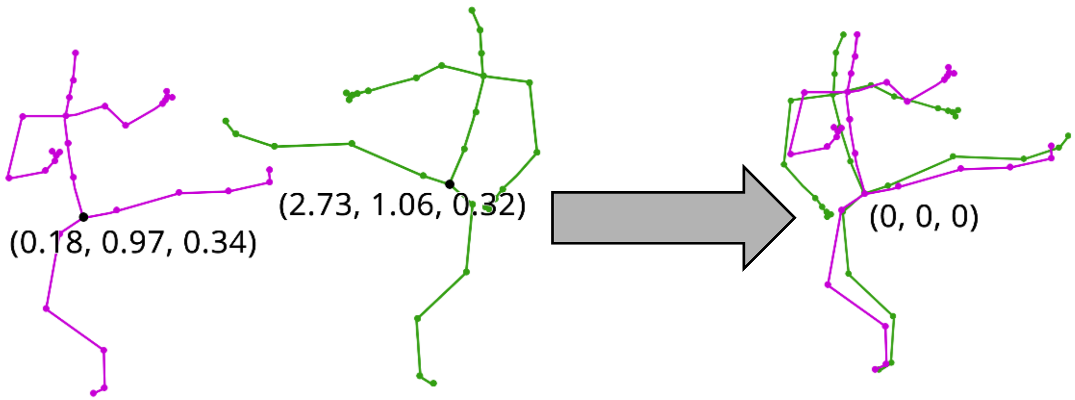
1) Normalizing motion data (optional context-dependent step)
2) Transforming normalized data into a 2D motion image
3) Extracting a 4,096D feature from the image using a DCNN



4.5 Feature Extraction – Normalization

Feature extraction steps

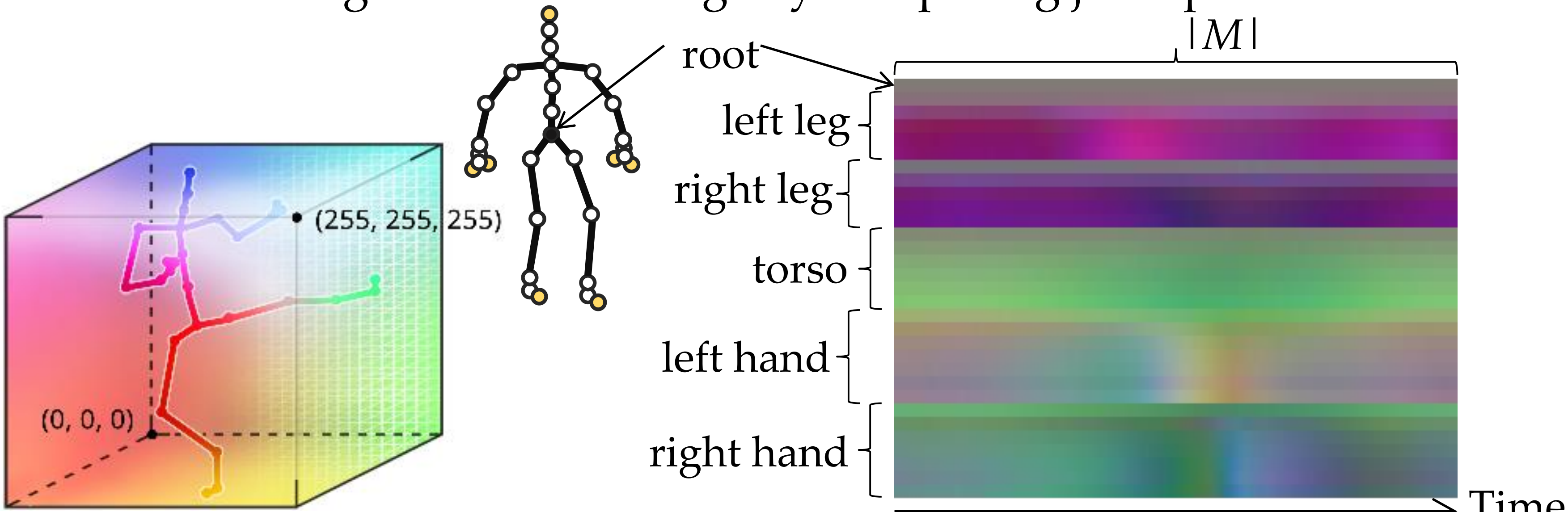
1) Normalizing motion data
   - Optional step – its utilization depends on a target application
   - Normalizing each pose independently vs. conditionally
   - E.g., position, orientation, and skeleton-size normalization in each pose independently is suitable for classifying daily activities



4.5 Feature Extraction – Visualization

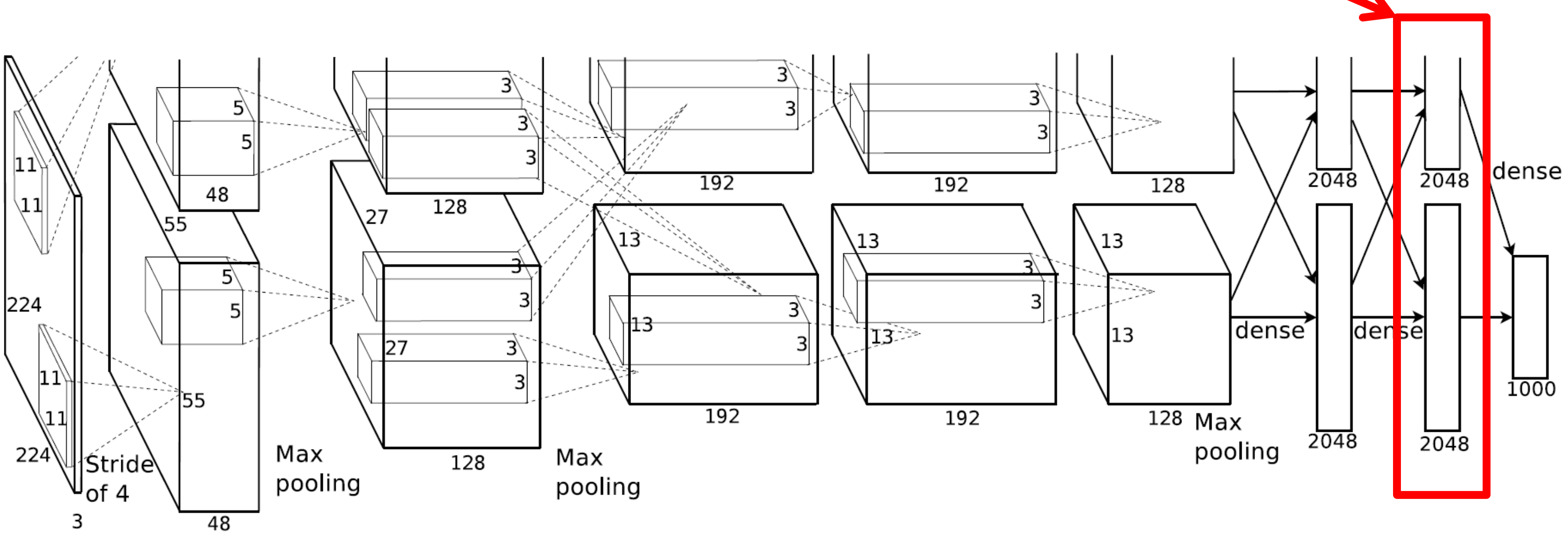
Feature extraction steps

2) Transforming data into a 2D motion image
   - Sizing an RGB cube to fit all possible poses of motion M
   - Fitting each motion pose into the center of the RGB cube to represent each joint position by a specific color
   - Building the motion image by composing joint-position colors



Feature extraction steps

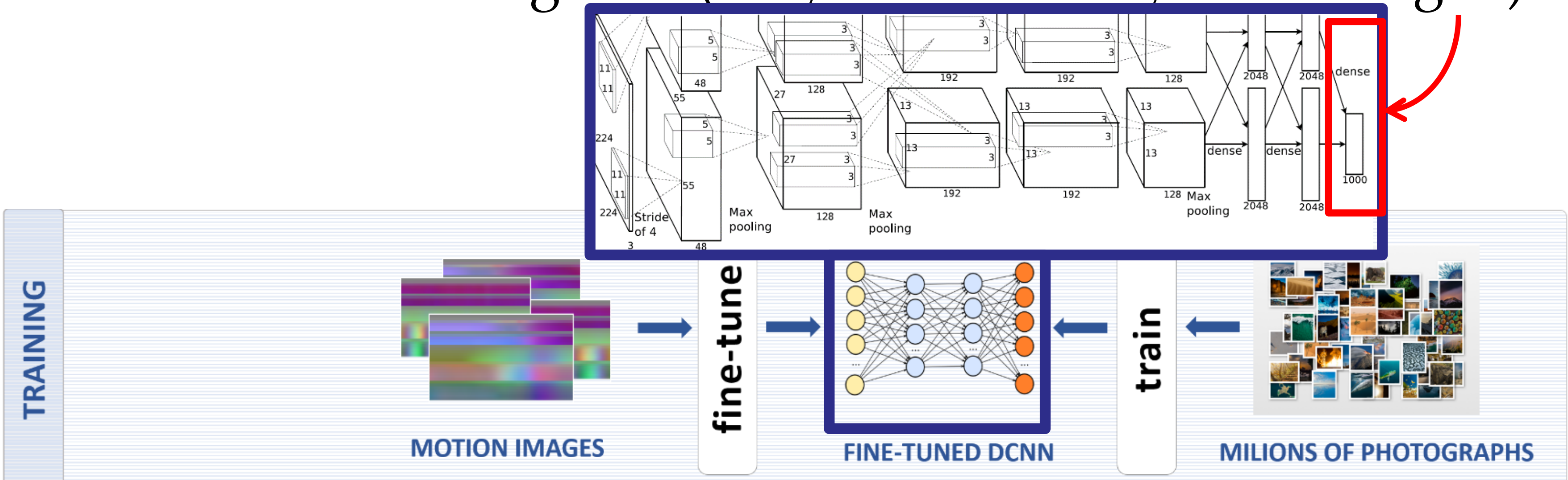
3) Extracting a 4,096D feature from the image using a CNN
   - CNN = AlexNet pretrained on 1M ImageNet photos categorized in 1,000 classes (e.g., green mamba, espresso, projector)
     - Optionally fine-tuned on the domain of motion images
   - 4,096D feature = output of the last hidden CNN layer



4.5 Increasing Accuracy of Features

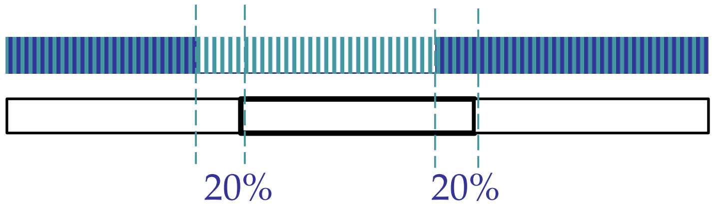
Fine-tuning the CNN ~ transferred learning

- Increases a descriptive power of the extracted features
- Utilizes a pre-trained CNN model, not-necessary originally trained on the same domain of images
- Requires additional domain-specific training images classified into categories (only last CNN layer is changed)

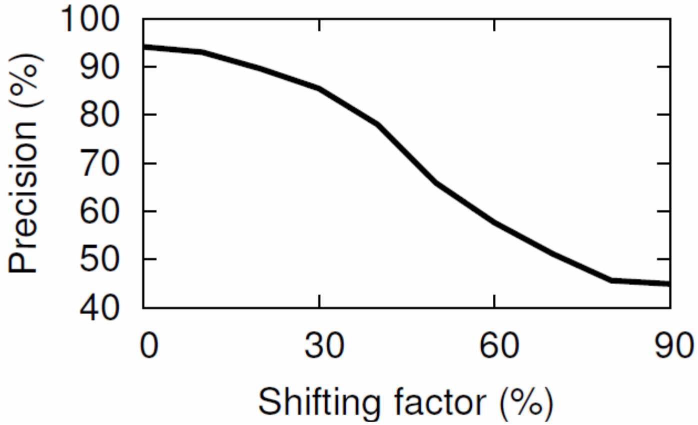


Elasticity property

- Motion-image similarity concept exhibits elasticity property
  - Classification accuracy decreases only slightly when up to 20% of motion content is misaligned (i.e., shifted)



20% misalignment w.r.t. segment size



  - Evaluated on the action recognition scenario using the 1NN classifier on a dataset of 1,464 HDM05 motions divided into 15 categories

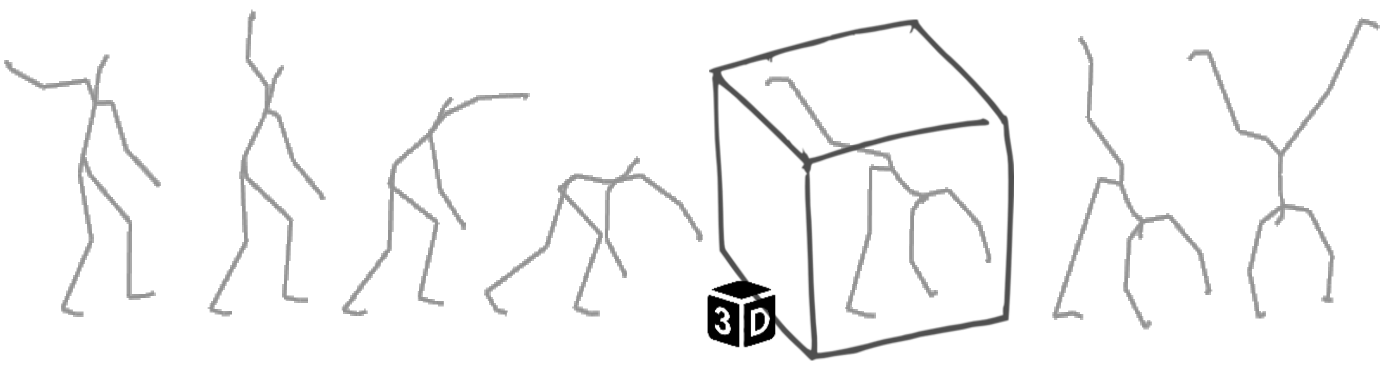
Summary of the motion-image similarity concept

- Suitable for motions in order of seconds (e.g., gait cycles)
  - Each motion image resized to 227x227 pixels for the DCNN
  - 227 pixels in time dimension correspond to the motion of ~2 seconds, when considering the frame rate of 120Hz
- Feature extraction time of ~25ms using a GPU impl.
- Advantages:
  - Utilizing a pre-trained CNN does not require large amounts of training data and training time
  - Combination of advantages of machine-learning techniques and distance-based methods
  - Even motions of categories that have not been available during the training phase are well clustered

Advantages/disadvantages of the CNN-based and LSTM-based similarity concepts

	CNN-BASED	LSTM-BASED
Accuracy (descriptive power of features)		
Volume of training data		
Input data preprocessing		
Length of motions		
Feature-size flexibility		
Complexity of network parametrization		

5 Classification of Segmented Motions



5.1 Action Classification

Action classification – the problem of identifying a single class (category) to which a query movement action belongs, on the basis of a training set of already categorized motions

- Sometimes referred to as action recognition

5.1 Action Classification

Knowledge base

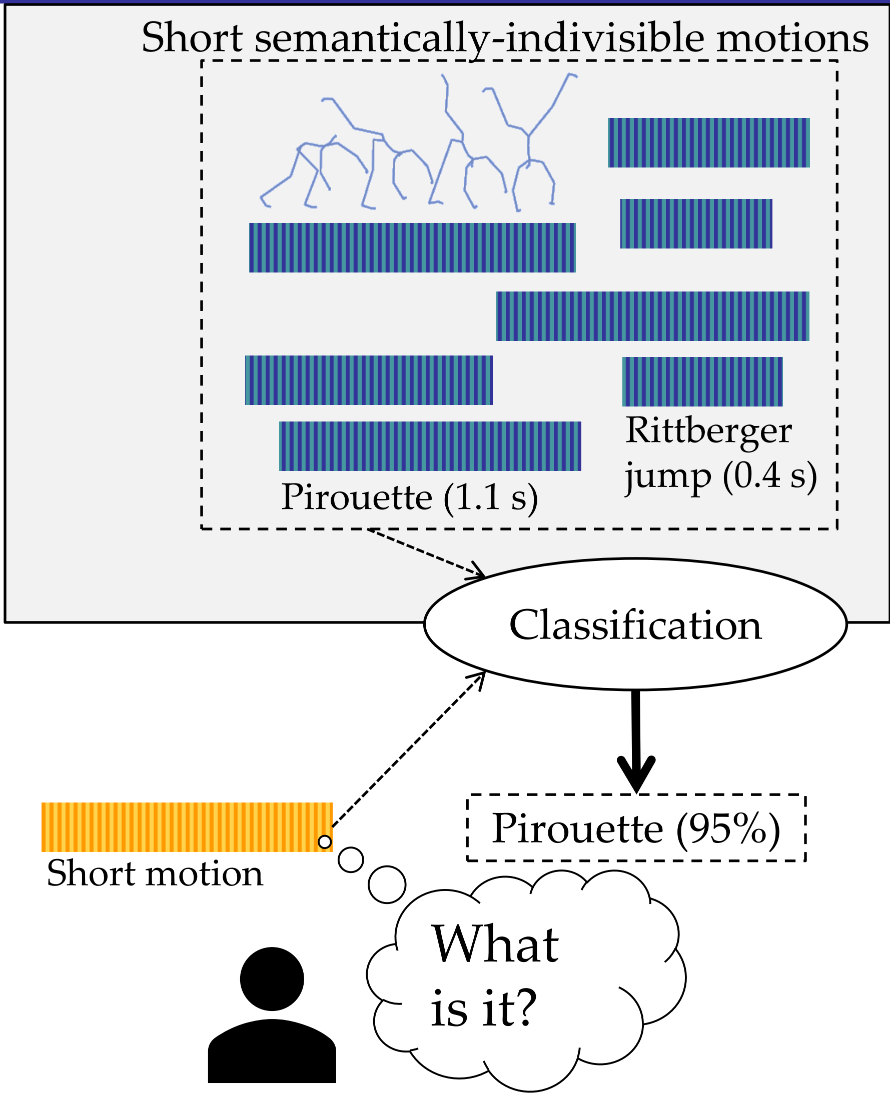
- Collection of labeled short actions ~ training data

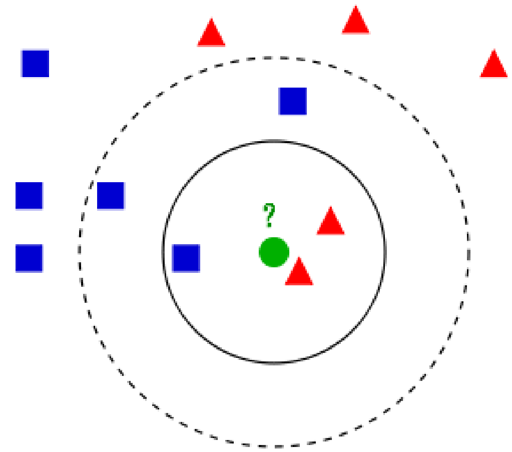
Input

- Unlabeled short action ~ query action

Output

- Estimated class of the query
- Probability of the query action being a member of each of the possible classes



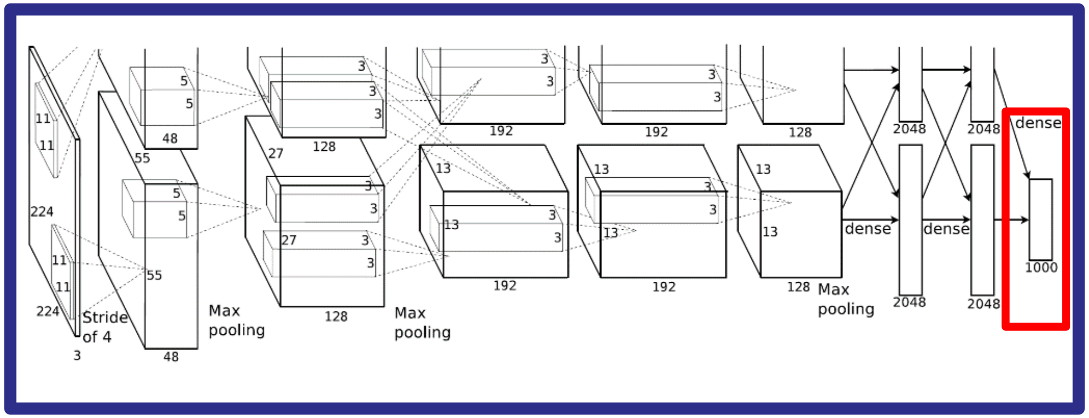


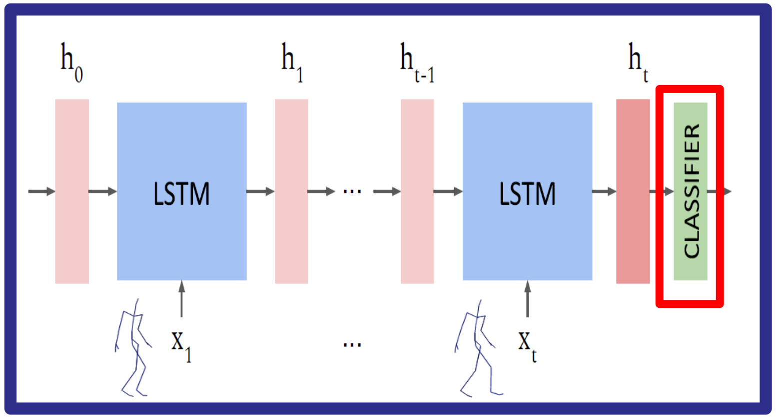
Action recognition approaches

- *k*-nearest-neighbor (*k*NN) classifiers
  - Require an effective similarity model (features + distance function)
  - Search for the *k* most similar actions with respect to the query
  - Rank the retrieved actions to estimate the query class (probability)
- Machine-learning (ML) classifiers
  - Learn the representation of classes from the provided training data
  - Query action is directly classified (usually in constant time)
  - Many approaches – support vector machines, decision trees, Bayesian networks, artificial neural networks

Neural-network-based classifiers

- Suitable architectures:
  - Convolutional (CNN) or recurrent (RNN) neural networks
- Training a network with categorized actions
  - (Re)Training is time-consuming
  - Network parameters are updated by processing each action
- Classifying an action without change of parameters

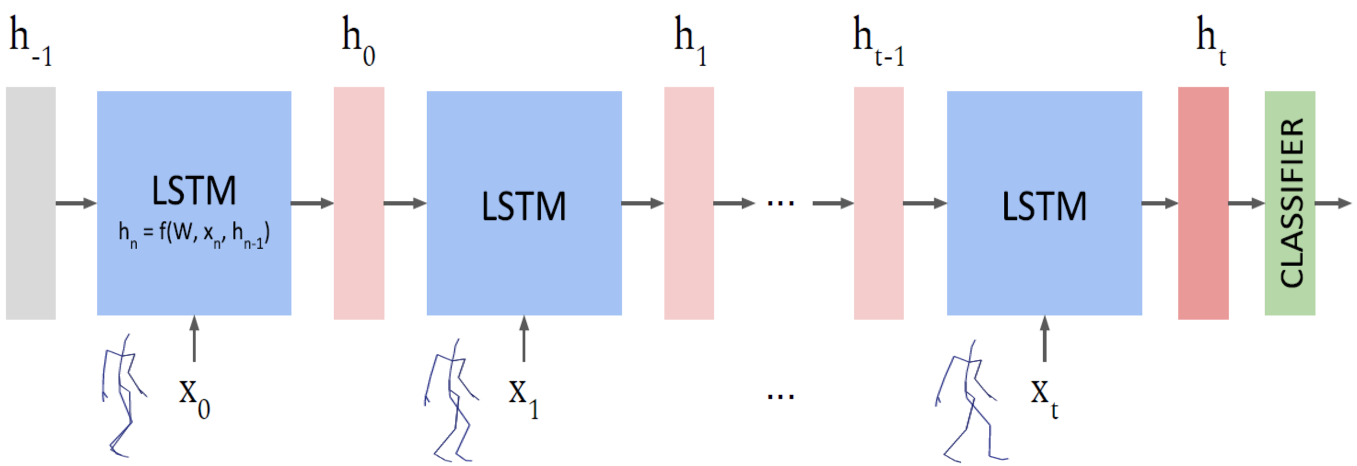




5.2 LSTM-Based Classifier

LSTM-based classifier (1kLSTM)

- Size of each state is set to 1,024 dimensions
- Classifier maps the last hidden state h_t into 122 categories



5.3 1NN-Based Classification

ACM MM 2018 Korea

1NN classification

- Searching for the nearest neighbor based on the motion similarity
- Class of the nearest neighbor considered as class of the query

JUMP class

feature vectors

<…, 0.53, 10.8, 4.64, …>

<…, 0.12, 8.60, 1.99, …>

Query action

feature vector

<…, 0.93, 10.1, 2.43, …>

1. **8.7 JUMP**
2. 10.9 KICK
3. 13.2 KICK
4. 14.3 KICK

⇒ JUMP (100%)

KICK class

feature vectors

<…, 8.93, 10.1, 2.43, …>

<…, 7.42, 7.14, 2.27, …>

<…, 3.93, 6.26, 3.41, …>

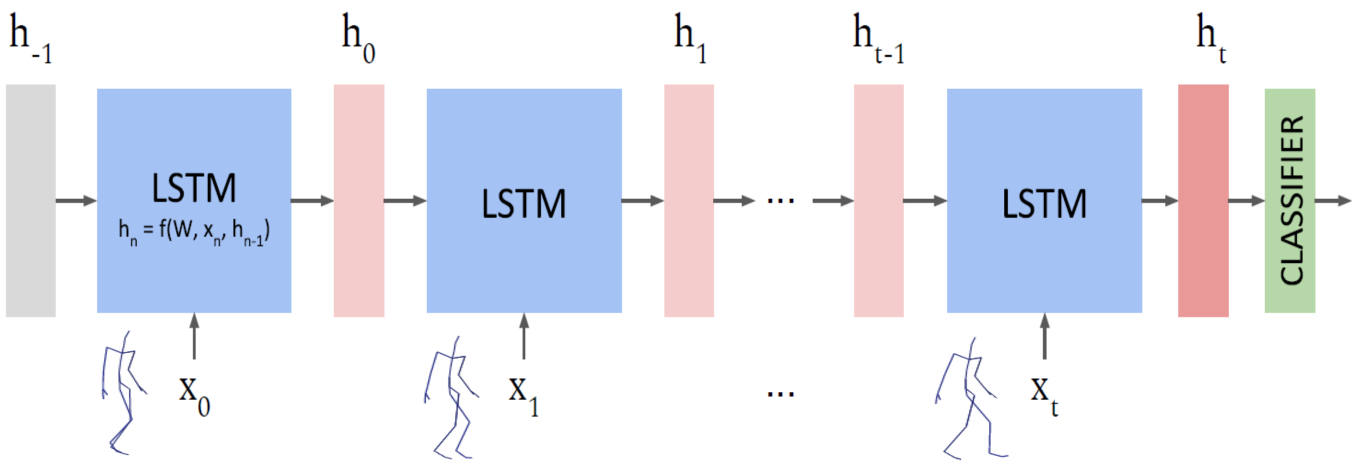
5.3 LSTM-Based 1NN Classifier

LSTM-based similarity concept

- The last hidden state h_t of 1,024 dimensions used as the action feature ~ 1kLSTM features
- The features of actions compared by the Euclidean function

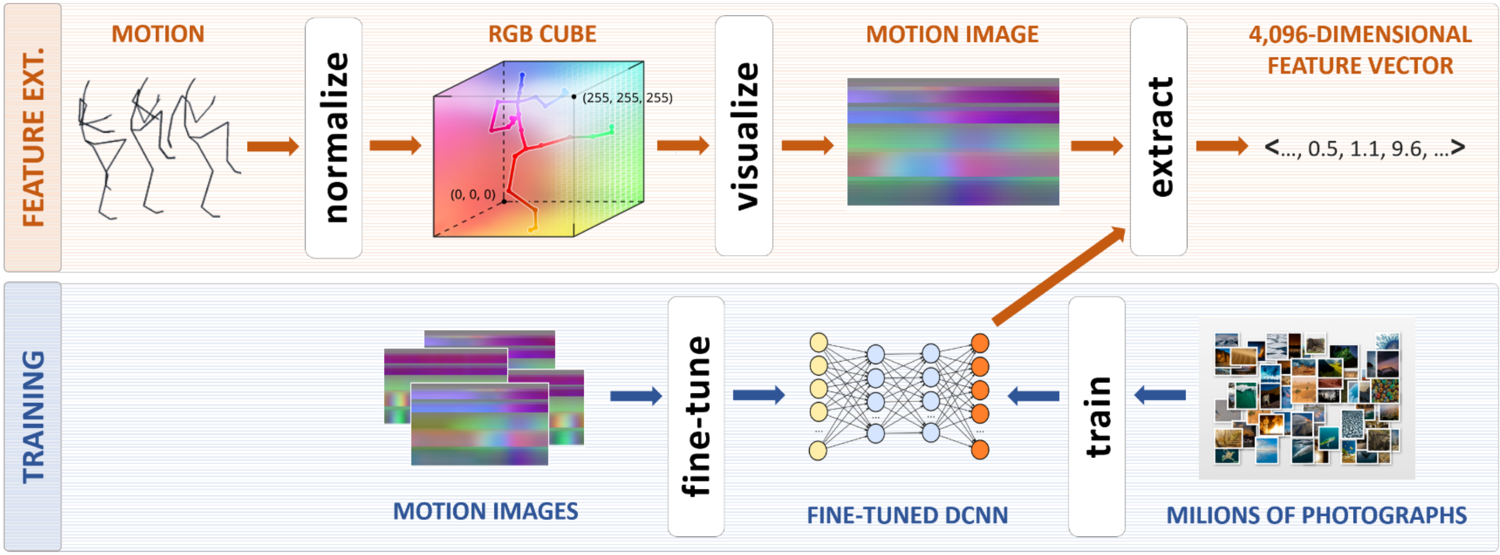
1NN classifier on 1kLSTM features

- 1NN classification using the 1kLSTM features



Motion-image 1NN classifier (1NN on 4kMI)

- 1NN classifier
- Similarity comparison:
  - Deep 4,096D features compared by the Euclidean distance function



[Sedmidubsky et al.: Effective and efficient similarity searching in motion capture data. Multimedia Tools and Applications, 2018]

5.3 *k*NN-Based Classification

1NN classification

- Problems – relying on the nearest neighbor only

*k*NN classification

- Possible design – considering the output class as the class with the highest number of occurrences within *k* results
  - If more candidates exist, take that with the minimum distance
- Problems:
  - When *k* is higher than the count of available class samples
  - Similarities of neighbors are not considered
  - Example: query action of the jump class

k=4:
1. 8.7 JUMP
2. 10.9 KICK
3. 13.2 KICK
4. 14.3 KICK

⇒ KICK (75%)
⇒ JUMP (25%)

5.3 *k*NN-Based Classification

ACM MM 2018 Korea

Weighted-distance *k*NN classifier (*k*NN_WD)

- Considering not only the number of votes but also the similarity of neighbors
  - Normalizing the neighbor distance with respect to the *k*-th neighbor
    - Effective when distances of nearest neighbors vary across classes
  - Computing class relevance by summing relevance of class neighbors (1 – normalized distance)
- Example scenario – query action belonging to the jump class

Original distances	Normalized distances	Relevance of neighbors	Relevance of classes	
1. 8.7 JUMP	1. 0.55 JUMP	1. 0.45 JUMP	0.45 JUMP	⇒ JUMP (45%)
2. 10.9 KICK	2. 0.69 KICK	2. 0.31 KICK	0.56 KICK	⇒ KICK (55%)
3. 13.2 KICK	3. 0.84 KICK	3. 0.16 KICK		
4. 14.3 KICK	4. 0.91 KICK	4. 0.09 KICK		

5.3 *k*NN-Based Classification

ACM MM 2018 Korea

Training-class-sizes *k*NN classifier (*k*NN_TCS)

- *k*NN_WD + considering also the count of class samples
  - Class relevance additionally modified by the square root of ratio between the number of class samples being among the *k*-nearest neighbors and the number of available training samples of that class
- Example scenario:
  - Knowledge base – **10** samples in kick class, **1** sample in jump class
  - Query – action belonging to the jump class

Original distances
1. 8.7 JUMP
2. 10.9 KICK
3. 13.2 KICK
4. 14.3 KICK

…

Relevance of classes

0.45 JUMP $\leftarrow \sqrt{\frac{1}{1}}$

0.56 KICK $\leftarrow \sqrt{\frac{3}{10}}$

Relevance modified

0.45 JUMP $\Rightarrow$ JUMP (59%)

0.31 KICK $\Rightarrow$ KICK (41%)

☺

5.4 Confusion-Based Classifier

Motivation

- 1NN classifier: ~87%
- *k*NN_WD/*k*NN_TCS classifier: <87%
- *k*NN_TCS "*benevolent*" classifier: ~95%

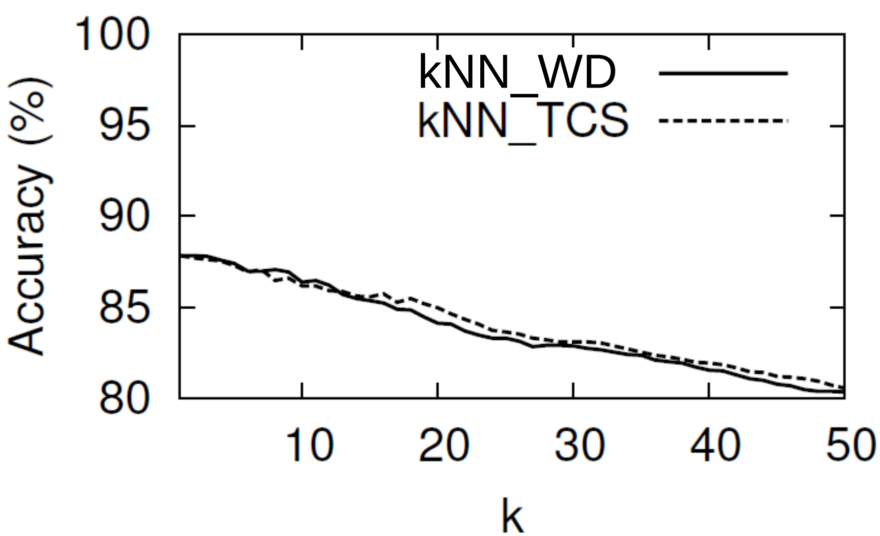
_k_NN_TCS

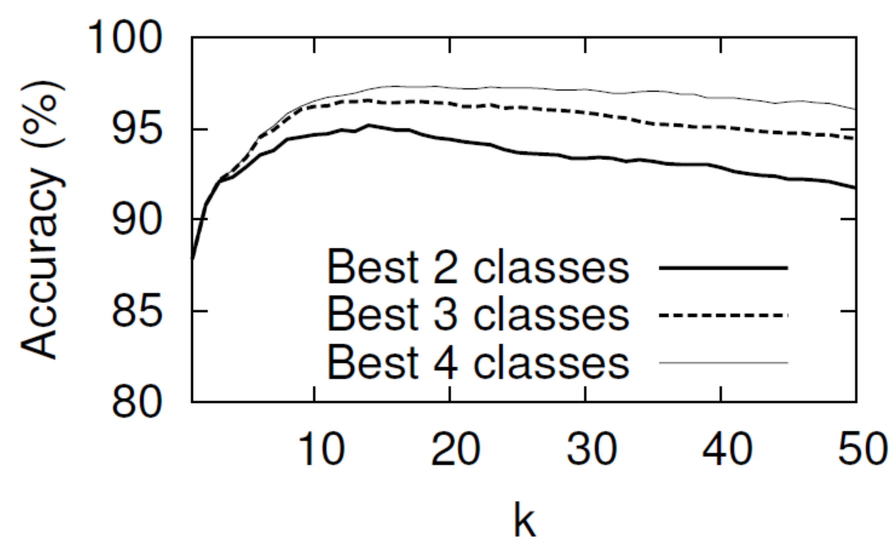
1. 8.7 KICK
2. 10.9 JUMP
3. 13.2 KICK
4. 14.3 KICK

⇒ **KICK (55%)**

⇒ **JUMP (30%)**

benevolent





5.4 Confusion-Based Classifier

Idea

- Use kNN_TCS classif. to determine the 2 most ranked classes
- Re-rank the k-nearest neighbors based on additional sim. functions that well separate that 2 most ranked classes

Training phase – additional similarity functions

- Learn a class confusion matrix *cm* (of size *#classes* x *#classes*) for each of n additional similarity functions
  - $cm^i[C_1, C_2] \in [0,1]$ – confusion of classes C_1 and C_2 based on the i-th similarity function ($i \in [1, n]$)
    - $cm^i[C_1, C_2] = 0$ indicates that the i-th function perfectly separates the motions of classes C_1 and C_2; with an increasing value, the separability decreases
  - $md^i[C_1, C_2] \in \mathbf{R}$ – maximum distance between motions of classes C_1 and C_2, with respect to the i-th similarity function

5.4 Confusion-Based Classifier

ACM MM 2018 Korea

Classification phase

1) Identifying the two most ranked classes
   – Utilizing the *k*NN_TCS classifier
2) Weighting similarity functions
   – Considering only the function(s) with the least confusability
3) Re-ranking and classifying neighbors
   – Aggregating weighted distances between the query and each neighbor
   – Re-ranking the neighbors by the computed distances
   – Outputting the class of the re-ranked nearest neighbor

5.4 Confusion-Based Classifier

Classification phase

1) Identifying the most ranked classes C_1 and C_2

kNN_TCS

1. 8.7 KICK
2. 10.9 JUMP
3. 13.2 KICK
4. 14.3 KICK
5. 14.4 JUMP
6. 14.8 JUMP
7. 16.2 PUNCH

⇒ **KICK (55%)** ← C_1 – the most ranked class
⇒ **JUMP (30%)** ← C_2 – the second most ranked class
⇒ PUNCH (15%)

5.4 Confusion-Based Classifier

Classification phase

2) Weighting similarity functions sim^i $(i \in [1, n])$

- Obtaining the minimum confusability *minConf*:

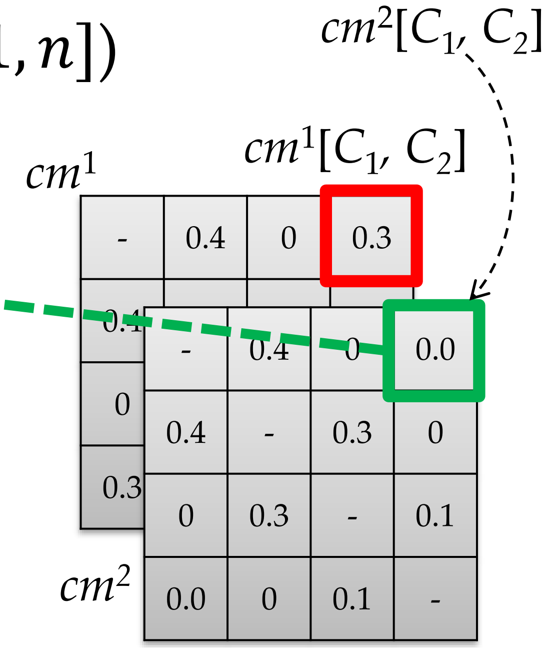
$$minConf = \min_{i \in [1,n]} \{cm^i_{C_1, C_2}\}$$

- Weighting additional similarity functions:

$$w^i = \begin{cases} 0 & cm^i_{C_1,C_2} > minConf \\ (1 - minConf)^3 & cm^i_{C_1,C_2} = minConf \end{cases}$$

- Weighting motion-image similarity function (*orig*):

$$w^{orig} = max\left\{(1 - cm^{orig}_{C_1,C_2})^3, 1 - (1 - minConf)^3\right\}$$



Classification phase

3) Re-ranking and classifying neighbors
   - Weighted distance is normalized based on the localized class-pairwise maximum distance

$$rerank(Q,M) = w^{orig} \cdot sim^{orig}(Q,M)/md^{orig}_{C_1,C_2} + \sum_{i=1}^{n} w^{i} \cdot sim^{i}(Q,M)/md^{i}_{C_1,C_2}$$

Q – query action to be classified

M – known labeled action

sim^i – *i*-th additional distance function

md^i – matrix of class-pairwise max. distances

	***k*NN_TCS**		**Re-ranked NNs**
1.	8.7 KICK	1.	**2.7 JUMP**
2.	10.9 JUMP	2.	4.4 JUMP
3.	13.2 KICK	3.	4.8 JUMP
4.	14.3 KICK	4.	8.9 KICK
5.	14.4 JUMP	5.	9.2 KICK
6.	14.8 JUMP	6.	9.6 KICK
7.	16.2 PUNCH	7.	10.2 PUNCH

$\Rightarrow$ **KICK (55%)**
$\Rightarrow$ **JUMP (30%)**
$\Rightarrow$ PUNCH (15%)

$\Rightarrow$ **JUMP (100%)**

Additional 3 similarity functions

- Manhattan (L_1) distance comparing these features:
  - Joint trajectory length – 31D feature vector, where each dimension corresponds to the total trajectory length of the specific joint
  - Normalized joint trajectory length (~joint speed) – 31D feature vector corresponding to the previous feature where all dimensions are additionally divided by the length of the motion sequence
  - Maximum axis distance – 93D feature vector whose dimensions correspond to the maximum reachable coordinate separately in the $x/y/z$ axis of each joint

5.5 Classification Dataset

HDM05 dataset

- Acquired by Vicon (120 Hz sampling, 31 body joints)
- 5 actors, 102 long motion sequences, 68 minutes in total
- Ground truth – 2,328/2,345 short actions in 122/130 classes
  - Shortest and longest samples: 13 frames (0.1s) and 900 frames (7.5s)
  - Action classes corresponding to daily/exercising activities:
    - Clap with hands 5 times
    - Walk two steps, starting with left leg
    - Turn left
    - Frontal kick by left leg two times
    - Cartwheel, starting with left hand

    ⋮

5.5 Comparison of Classification Methods

- HDM05 dataset 2,328/2,345 samples in 122/130 classes
- 2-fold cross validation (50% of training data)
  - Only about 10 action samples per class for training on average

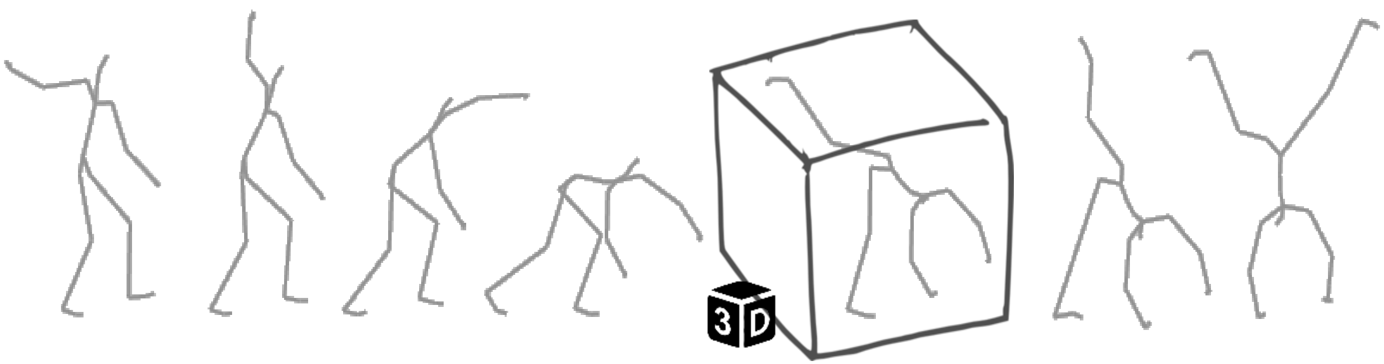
Method		Accuracy (%)	
		HDM-122	HDM-130
Related approach	Huang et al. (2016)	N/A	75.78
	Laraba et al. (2017)	N/A	83.33
	Li et al. (2018)	N/A	86.17
Presented approach	1NN on 4kMI (2017)	87.24	86.79
	1NN on 4kMIE (2017)	87.84	87.38
	Confusion-based 15NN_TCS on 4kMIE (2018)	89.09	88.78
	1NN on 1kLSTM (2018)	90.60	N/A
	1kLSTM classification (2018)	91.20	N/A

Advantages/disadvantages of the *k*NN and ML classifiers

	*k*NN-BASED	ML-BASED
Accuracy	🙂	🙂
Training time	🙂 ☹️	☹️
Adaptability to a changing knowledge base	🙂	☹️
Classification efficiency	😐	🙂

- Demo: http://disa.fi.muni.cz/mocap-demo-classification/

6 Processing Long and Unsegmented Motion Sequences



Long motions

- Semantically-**divisible** motions ~ sequence of actions
- Length – in order of minutes, hours, days, or even unlimited
- Database – typically a single long motion either pre-processed as a whole, or evaluated in the stream-based nature

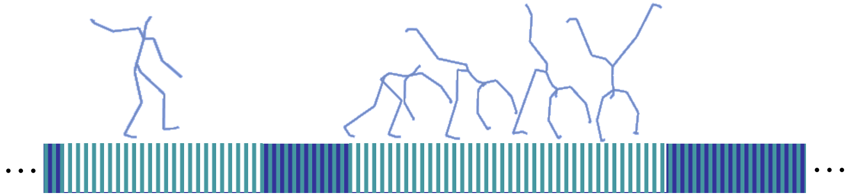
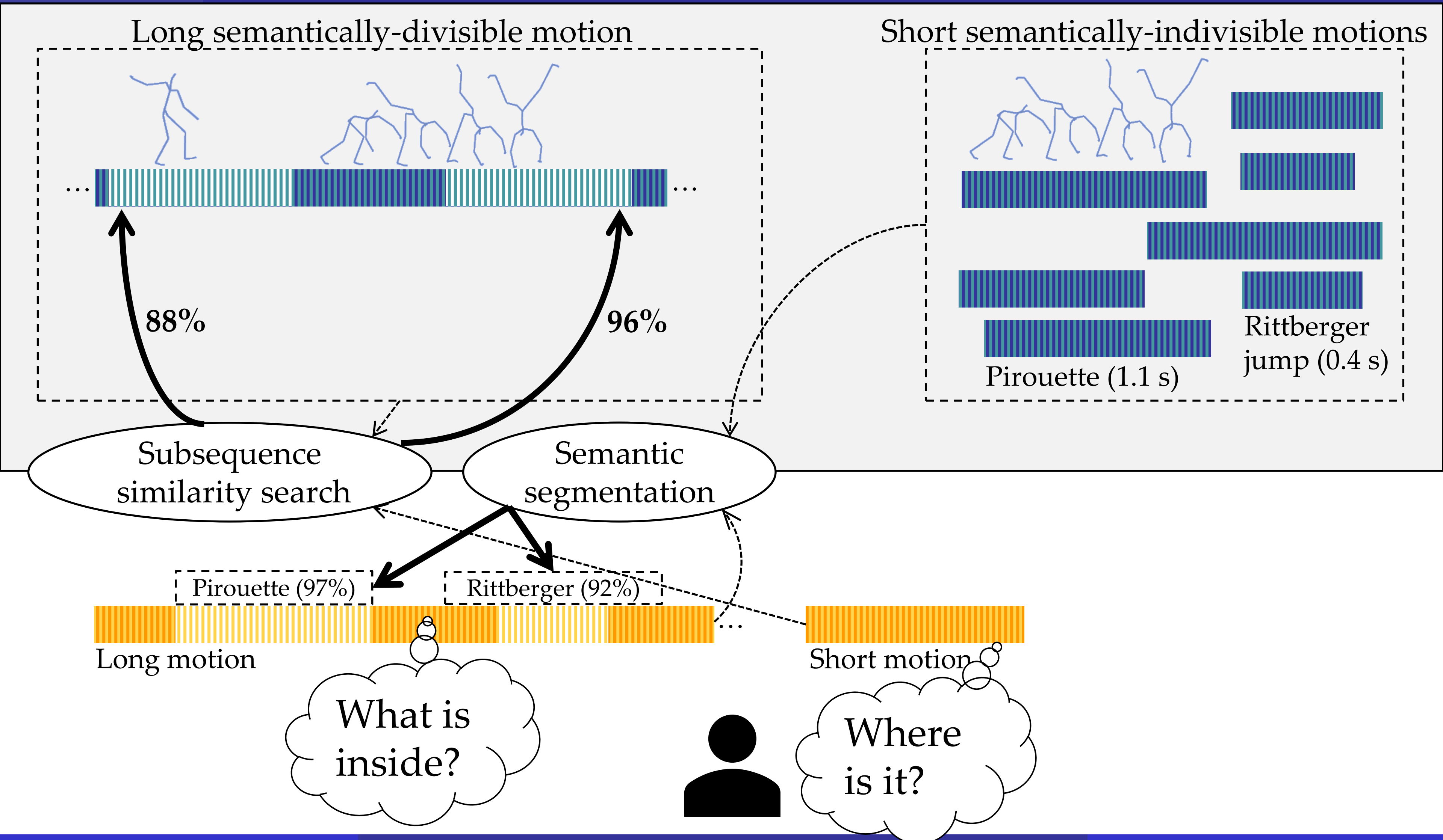


Figure skating performance (3 mins)

6.1 Processing Long Motions

ACM MM 2018 Korea



6.1 Processing Long Motions

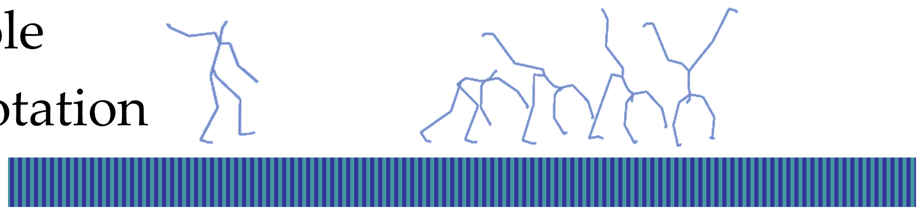
Operations

- Subsequence similarity search
- Semantic segmentation
  - Offline sequence annotation
  - Real-time event detection
- Other operations:
  - Mining frequent movement patterns
  - Prediction of actions

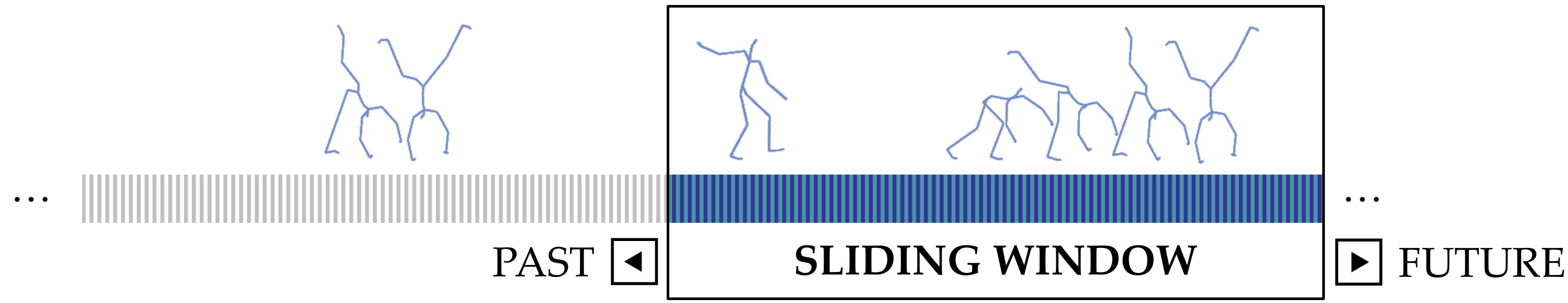
6.1 Processing Long Motions

Long-motion processing

- File-based processing:
  - The long motion is known in advance and can be stored and pre-processed offline as a whole
  - E.g., offline sequence annotation

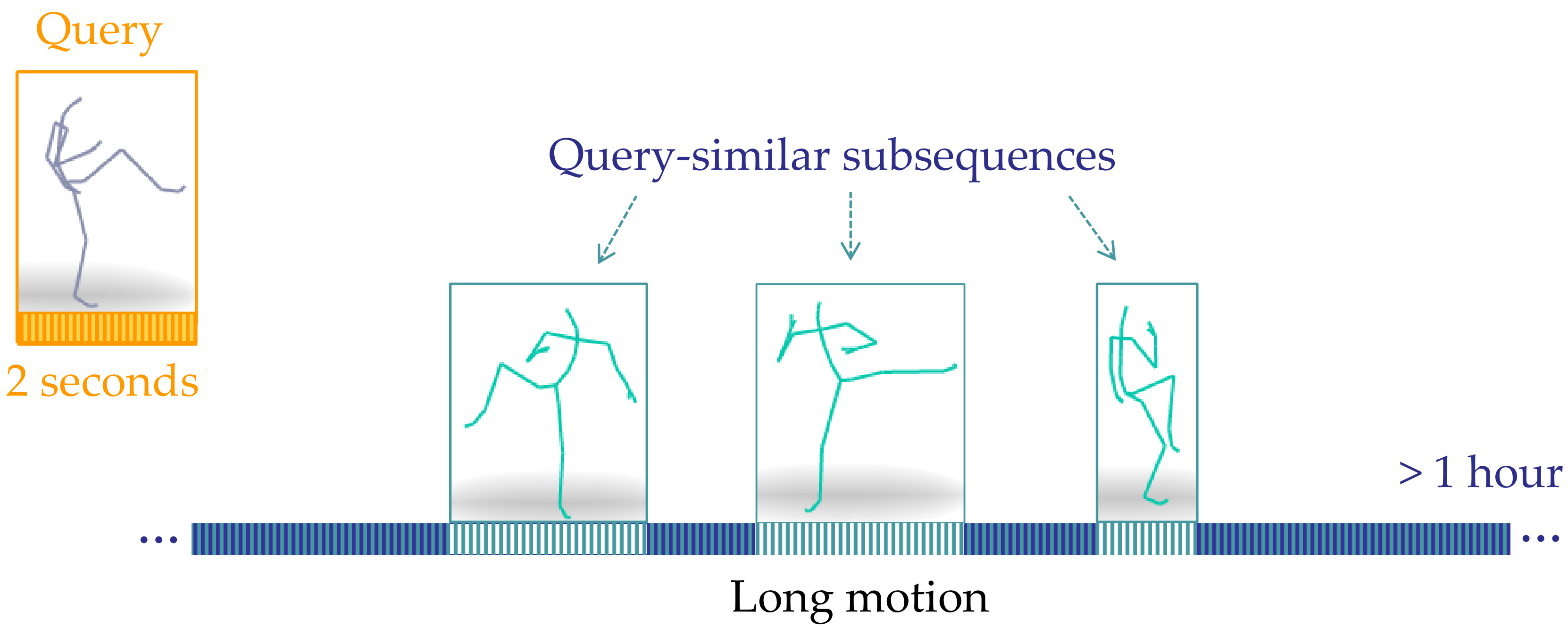


- Stream-based processing:
  - A limited part of the long motion is accessible at a given time
  - E.g., real-time event detection in data from surveillance cameras



Subsequence search

- An efficient mechanism for searching a long motion and localizing its parts that are similar to a short query sequence



6.2 Search Challenges

Problems

- Query can be potentially any motion sequence, usually limited in its length
  - E.g., semantic action such as kick or jump, its part or a transition in between any of these, but also any non-categorized motion
- Query-similar subsequences can potentially occur anywhere in a long sequence
- Length of query-similar subsequences needn't be exactly the same with respect to the query motion

=> efficient subsequence matching algorithm

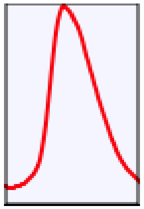
Subsequence matching in time series

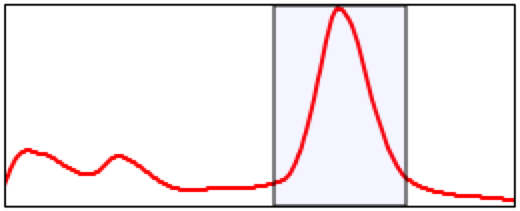
- Motion data can be perceived as a set of synchronized time series ~ a single multi-dimensional time series
  - E.g., a single time series for each joint and axis (*x*/*y*/*z*)
    => 31 joints · 3 = 93 time series

- Subsequence matching in time series data is a well-known problem for 1-dimensional time series

[Esling et al.: Time-series data mining. ACM Computing Surveys, 2012.]

[Rakthanmanon et al.: Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping. KDD 2012]



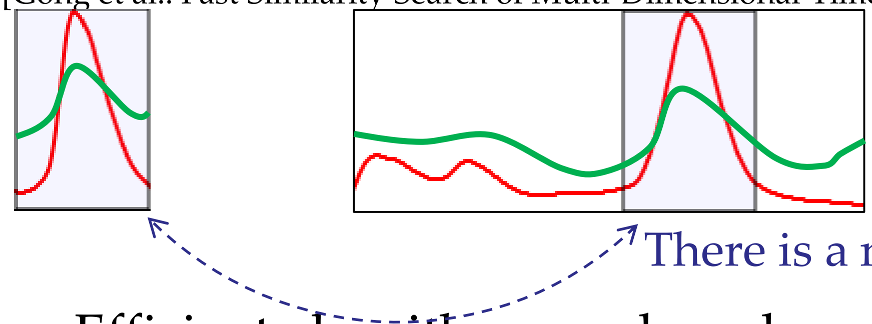


Subsequence matching in time series

- Subsequence matching in time series data also applied to multi-dimensional time series

  [Hu et al.: Time Series Classification under More Realistic Assumptions. ICDM, 2013.]

  [Gong et al.: Fast Similarity Search of Multi-Dimensional Time Series via Segment Rotation. DASFAA, 2015.]



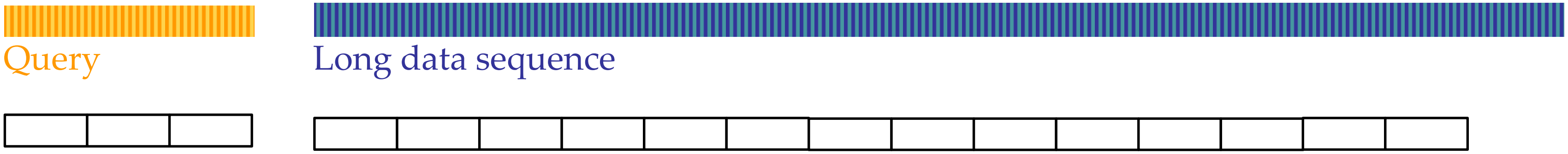
There is a need for an effective distance function

  – Efficient algorithms are based on distance functions that compare frame-based features

- Traditional time-series algorithms hardly applicable to motion-data domain due to the absence of distance functions working **effectively** on **frame-based features**

Subsequence matching in motion data

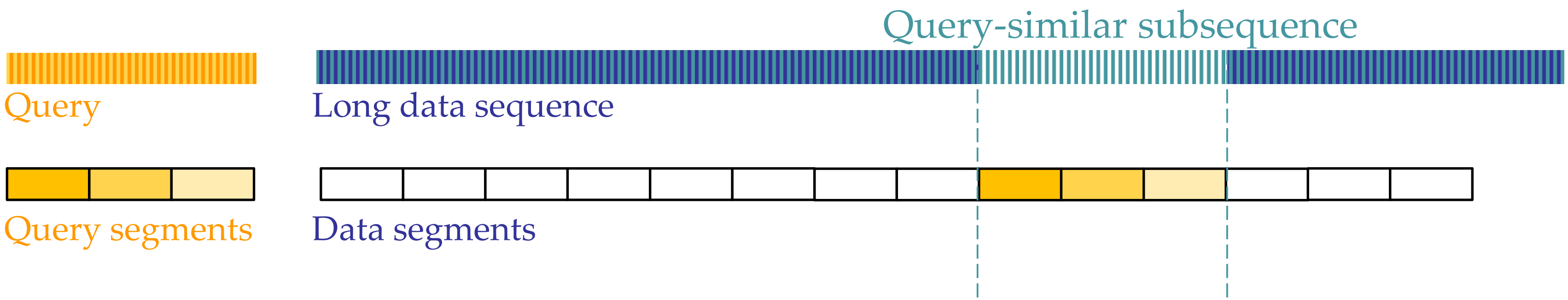
- Effective motion-based features are extracted from short motions => segmentation
- Partitioning the query and long motion sequence into parts – segments – to be meaningfully comparable



- Types of segmentation:
  - Overlapping/disjoint segments
  - Segments of a fixed/variable length
  - Unsupervised/supervised (semantic) segmentation

Subsequence matching in motion data

- Subsequence search = segmentation + retrieval algorithm
- Retrieval algorithm – searching for consecutive data segments that are similar to consecutive query segments



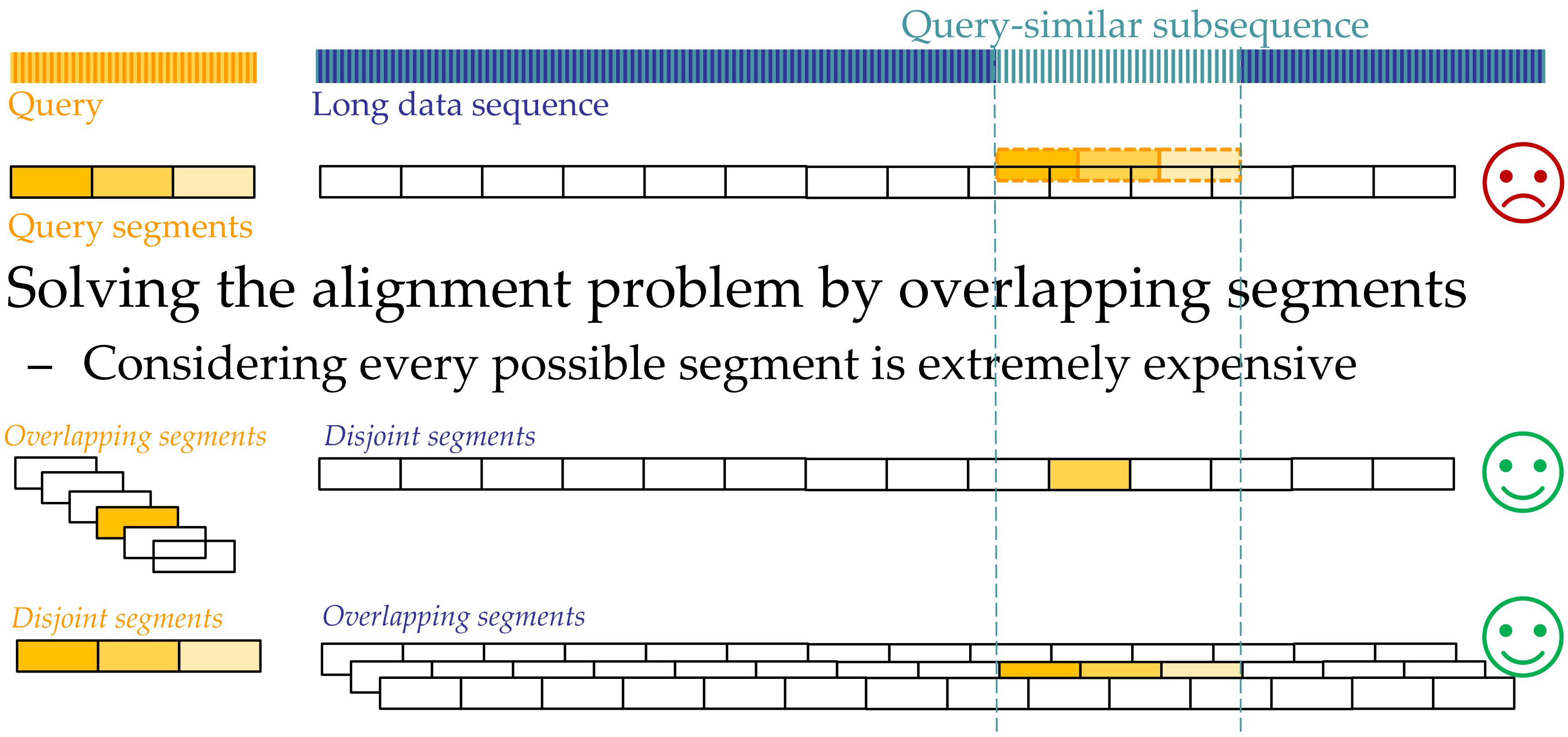
6.2 Alignment Problem

Alignment problem in subsequence matching

⇒ Detecting only "*selected*" segments => alignment problem

⇒ Solving the alignment problem by overlapping segments

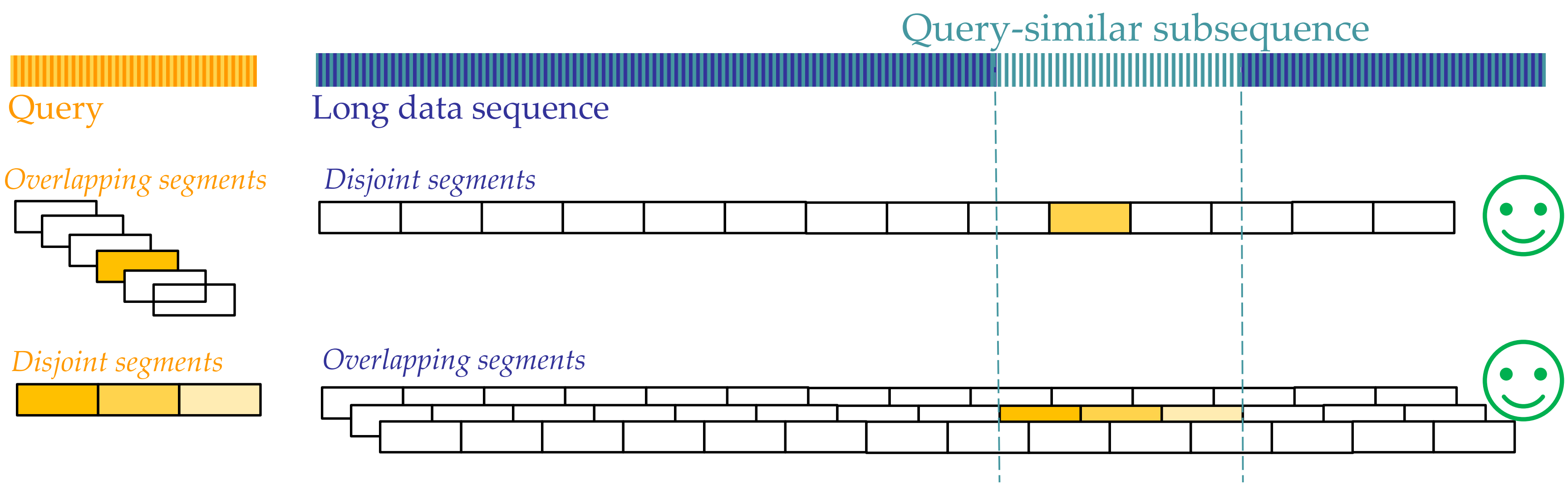
– Considering every possible segment is extremely expensive



6.2 Overlapping Segmentation

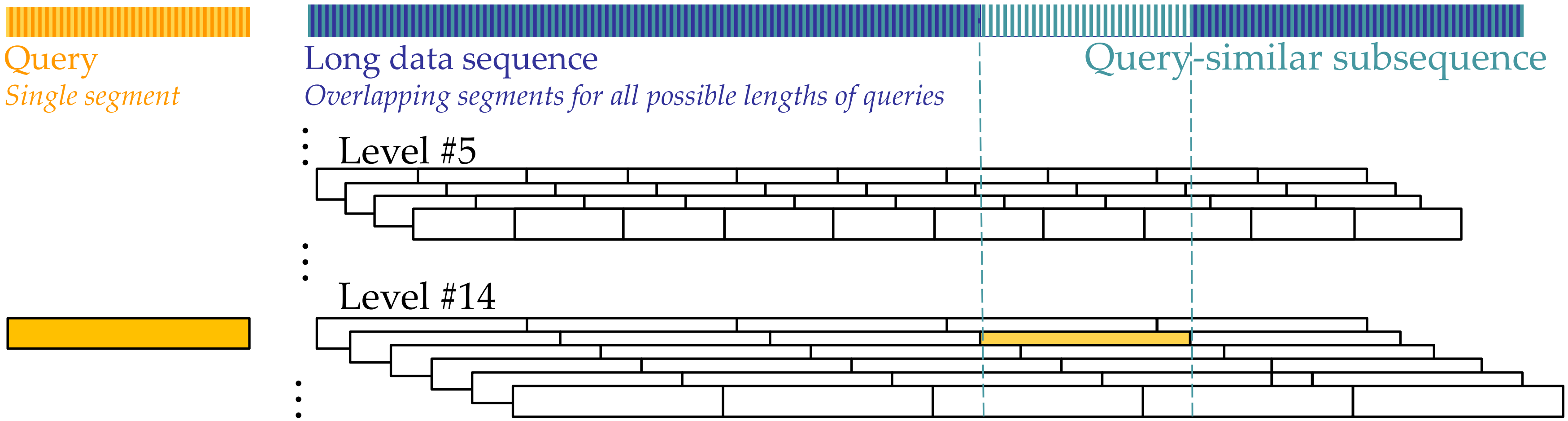
Partitioning both the query and data sequence

- ☺ Overlapping segments solve the alignment problem
- ☹ Longer queries have more query segments and are more expensive to evaluate
- ☹ Grouping relevant segments w.r.t. temporal information



Partitioning only the data sequence

- Solving the alignment problem by:
  - Considering a query as a single segment
  - Organizing overlapping data segments in multiple levels for different segment lengths
- ☺ Much easier retrieval – one query, no complex post-processing
- ☹ Segment level for each query length – a big number of data segments

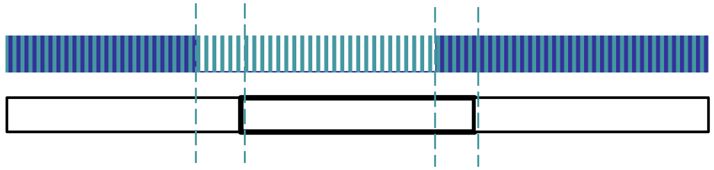


[Sedmidubsky et al.: Similarity Searching in Long Sequences of Motion Capture Data. SISAP 2016]

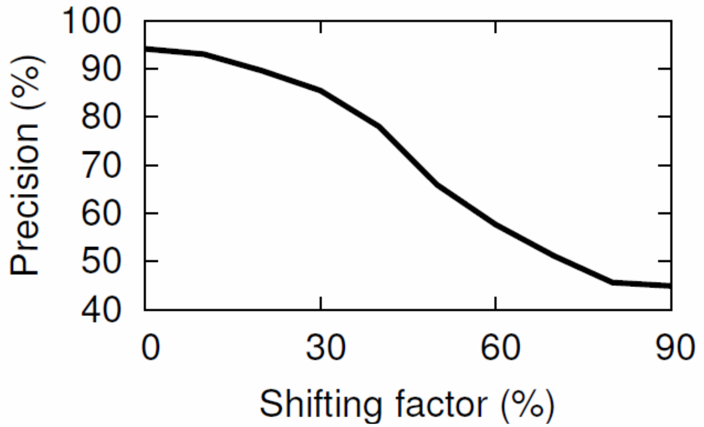
ACM MM 2018 Korea

Reducing the number of levels and segments

- Motion-image similarity concept exhibits elasticity property
  - Search accuracy decreases only slightly when up to 20% of segment content is misaligned (i.e., shifted)



20% misalignment w.r.t. segment size



Overlapping segments can be shifted by 5–25 % of their length (and not only by a single frame)

Levels can be generated only for the specific lengths of queries (and not for all the possible ones)

☺ The big number of segments can be dramatically reduced

6.2 Decreasing Number of Segments

Reducing the number of levels and segments

- Segment lengths and number of levels depend on
  - Query length limits (l^{min}, l^{max})
  - Elasticity of the similarity measure (quantified by $cf \in [0, 1]$)

- Segmentation example for elasticity $cf = 0.2 \sim 20\%$:

Query length limits [100, 500]

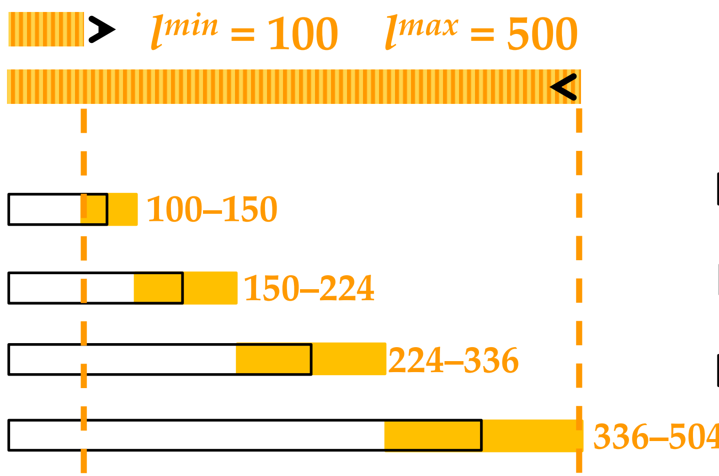
Segment levels

#**1** ($l_1 = 125$ *frames*)

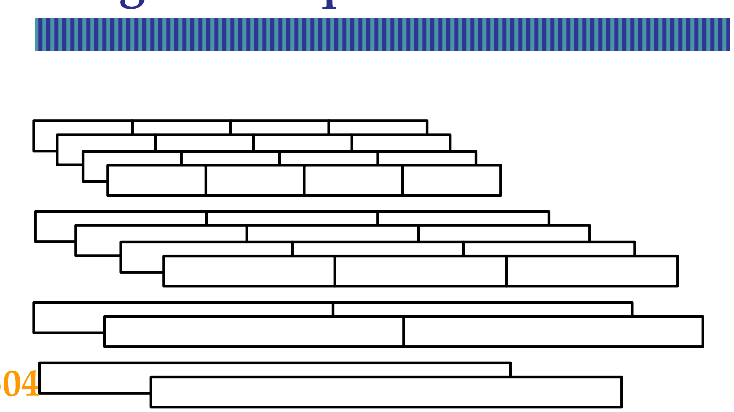
#**2** ($l_2 = 187$ *frames*)

#**3** ($l_3 = 280$ *frames*)

#**4** ($l_4 = 420$ *frames*)



Long data sequence

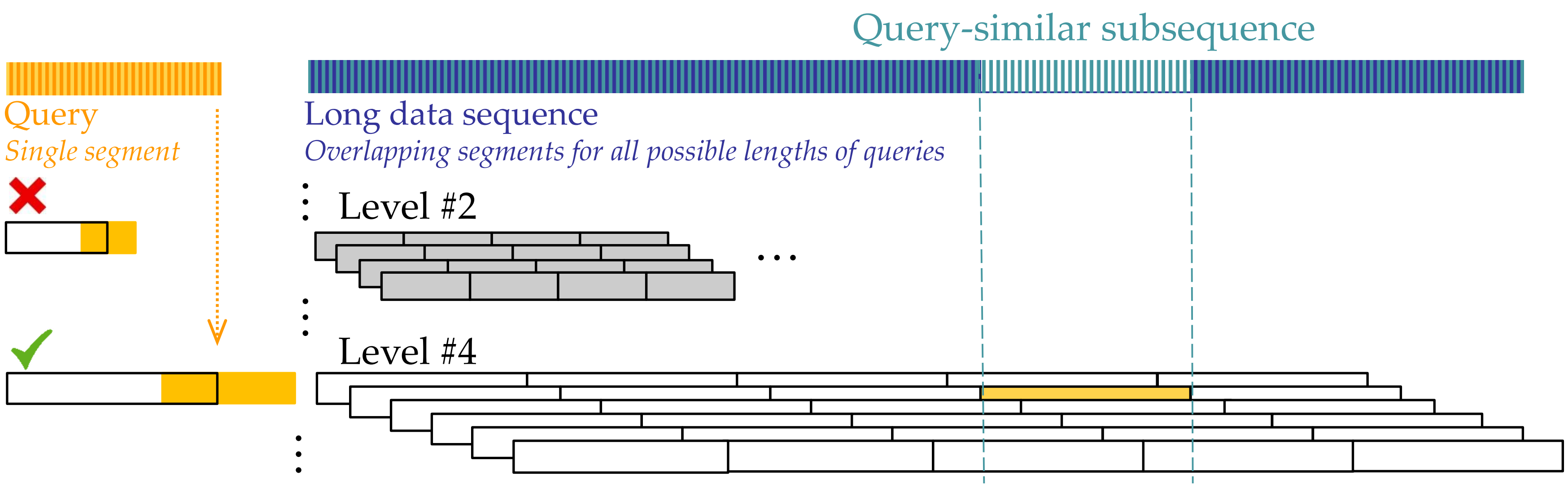


Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

Segment shift: $l_n * cf$

Searching within a multi-level segmentation

- Only a single query-relevant level considered for search
  - For arbitrary data subsequence of l^{min} < length < l^{max}, there exists a single segment that overlaps for at most $100 \cdot (1 - cf)$ [%]
- The *k* most similar segments presented as the query result



6.2 Query Evaluation Costs

Example:

- Data sequence of length 400,000 frames (120 Hz ~ 1 hour)
- Query length limits: $l^{min} = 100$ and $l^{max} = 500$ frames
- Example query length: 300 frames (120 Hz ~ 3 seconds)

	Total # of data segments	Data replication	Max # of comparisons
Baseline – overlap on query	4,000	1	800,000
Baseline – overlap on data	400,000	100	1,200,000
Multi-level segmentation – naïve	160,000,000	120,000	400,000
Multi-level segmentation	7,720	20	1,430

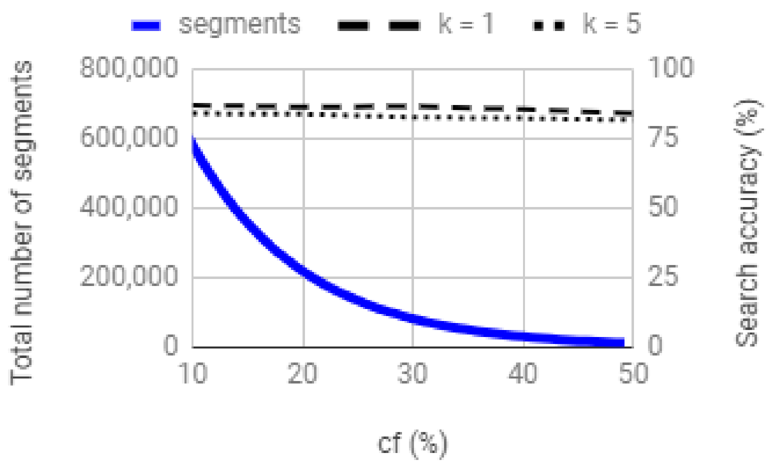
HDM05 – long motions

- 102 long sequences ~ 68 minutes in total
- Ground truth – 1,464 short subsequences in 15 categories (~queries)
  - Shortest and longest samples: 41 frames (0.3s) and 2,063 frames (17.2s)
  - Action classes corresponding to exercising activities:
    - Cartwheel
    - Exercise
    - Jump
    - Kick

    ⋮

Subsequence search evaluation

- Subsequence retrieval using *k*NN queries:
  - 1,464 ground-truth subsequences used as query objects
  - Retrieved subsequence is relevant if it overlaps with some ground-truth subsequence of the same class
  - $l^{min} = 41$ frames (0.3s), $l^{max} = 2{,}063$ frames (17.2s)
  - Different settings of elasticity $cf = \{10\%, 20\%, 30\%, 40\%, 50\%\}$

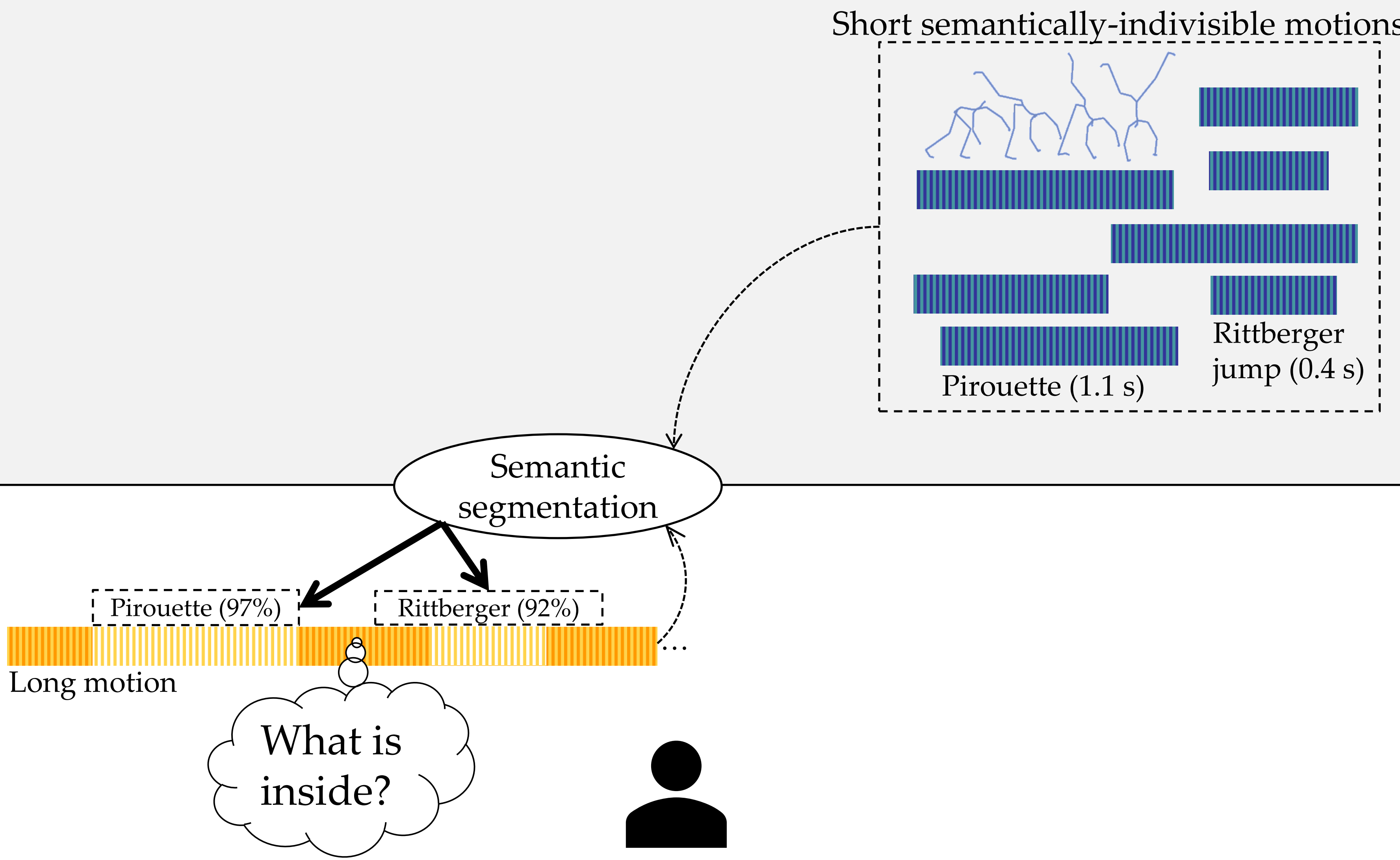


cf [%]	# of levels	Sequential scan [ms]
10	18	447
20	9	205
30	6	126
40	5	88
50	4	66

Summary

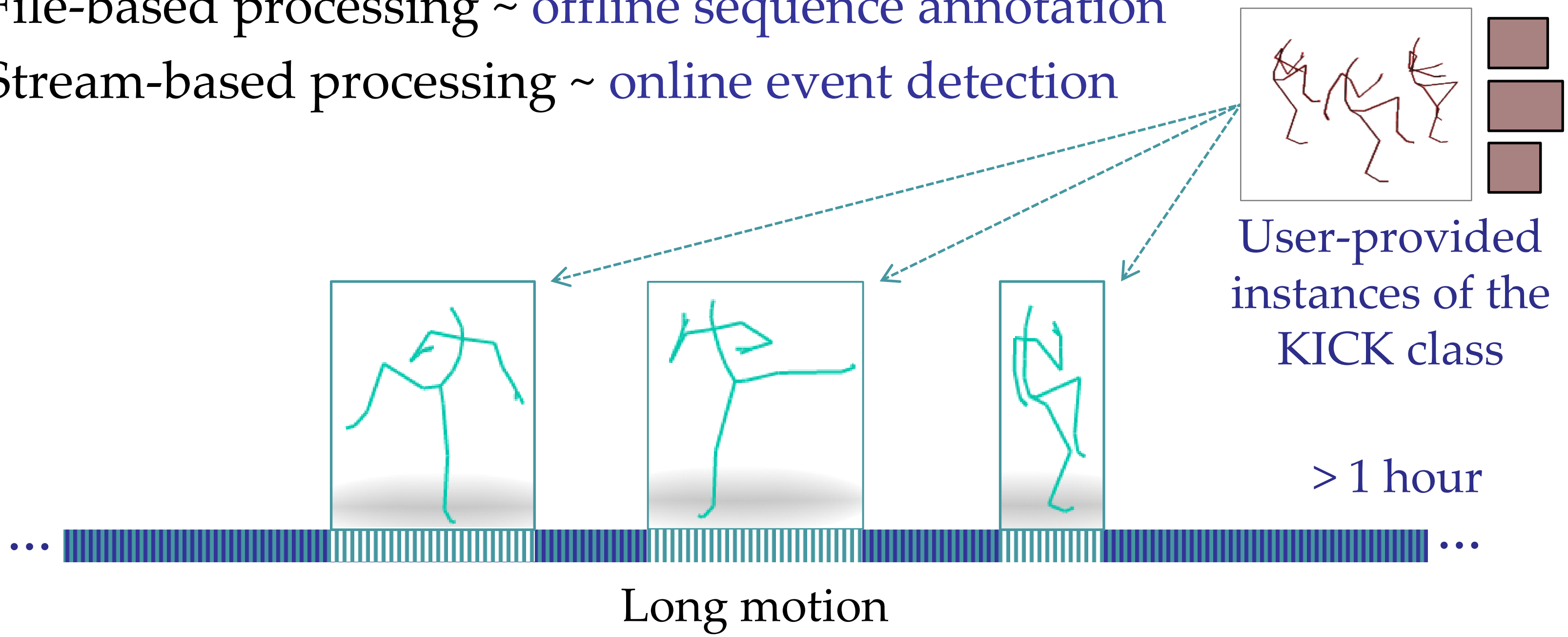
- Advanced subsequence matching in mocap data:
  - Query always considered as a single segment
  - The elasticity property of the motion-image similarity concept dramatically reduces the number of data segments
- Efficiency:
  - Searching the 68-minute sequence sequentially takes 205ms
  - Search times can further be decreased by roughly two orders of magnitude by indexing data segments at each level
    - Approximate search within a 121-day long data sequence in 1 second
- Demo: http://disa.fi.muni.cz/mocap-demo-classification/

6.3 Semantic Segmentation



Semantic segmentation

- An efficient mechanism for discovering actions within a long motion, based on a user-provided categorization
- Processing:
  - File-based processing ~ offline sequence annotation
  - Stream-based processing ~ online event detection



6.3 Semantic Segmentation

Challenges

- Beginnings and endings of actions are unknown
  - A more difficult problem than action classification
- In case of stream-based processing, only a small part of data is accessible and has to be processed in real time

Approaches

- Segment-based event detection
  [Elias et al.: A Real-Time Annotation of Motion Data Streams, ISM 2017]
- Frame-based semantic segmentation using a LSTM network
  - Offline-LSTM – offline sequence annotation
  - Online-LSTM – online event detection

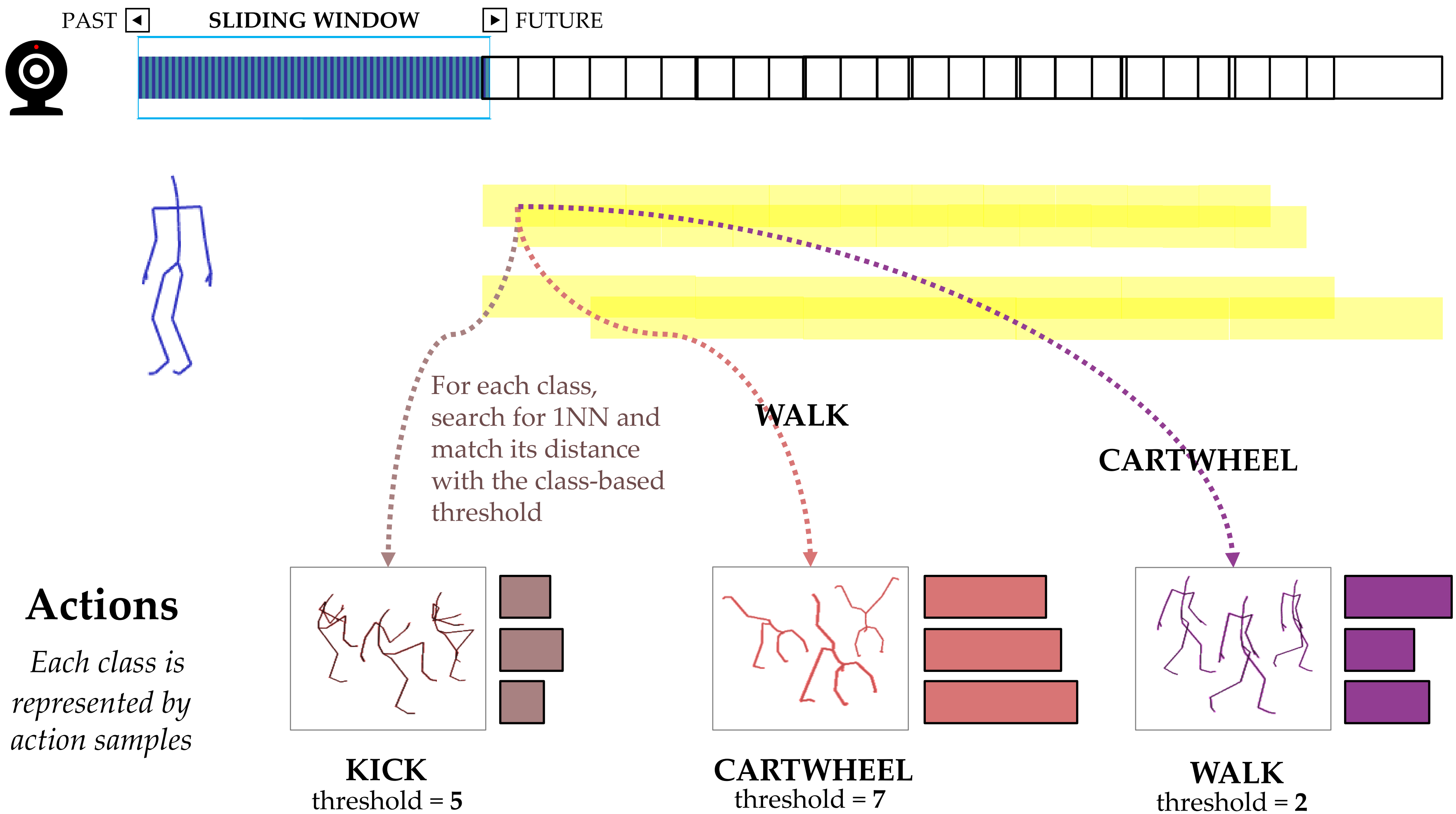
6.3 Segment-Based Event Detection

Segment-based matching

- Multi-level segmentation structure as in subsequence search
  - Segments detected in stream-based nature
- Each segment is matched against each action in each class
  - Matching based on motion-image similarity concept
  - If similarity between the segment and action is under a class-based threshold, the segment is assigned the action class
  - All the assigned segments are merged to obtain the overall semantic segmentation

6.3 Segment-Based Event Detection

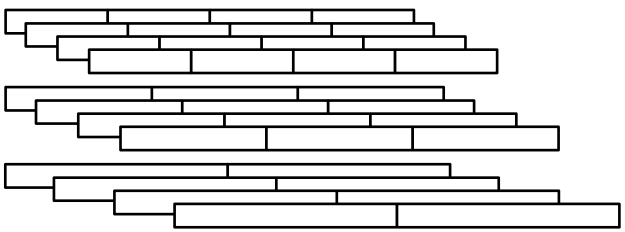
ACM MM 2018 Korea



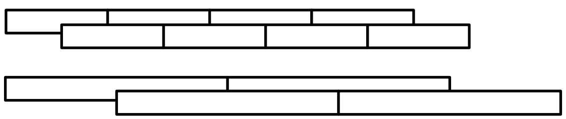
6.3 Segment-Based Event Detection

Segmentation

- Multi-level segmentation structure as in subsequence search
  - Versatility – the density of the segments is controlled by a user-specified parameter *cf*
  - The parameter denotes the number of levels and the size of shift (overlap) between consecutive segments



Dense segmentation
Produces more segments resulting in a more precise annotation but requires more processing power.

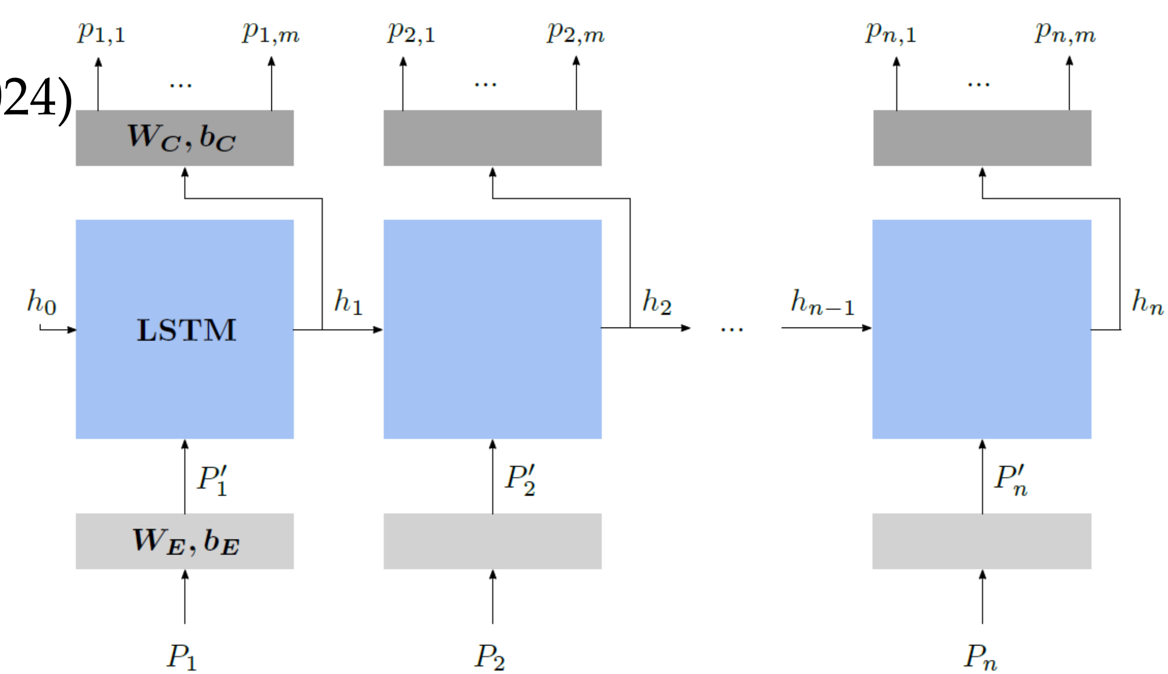


Sparse segmentation
Produces less segments but requires a more elastic similarity measure.

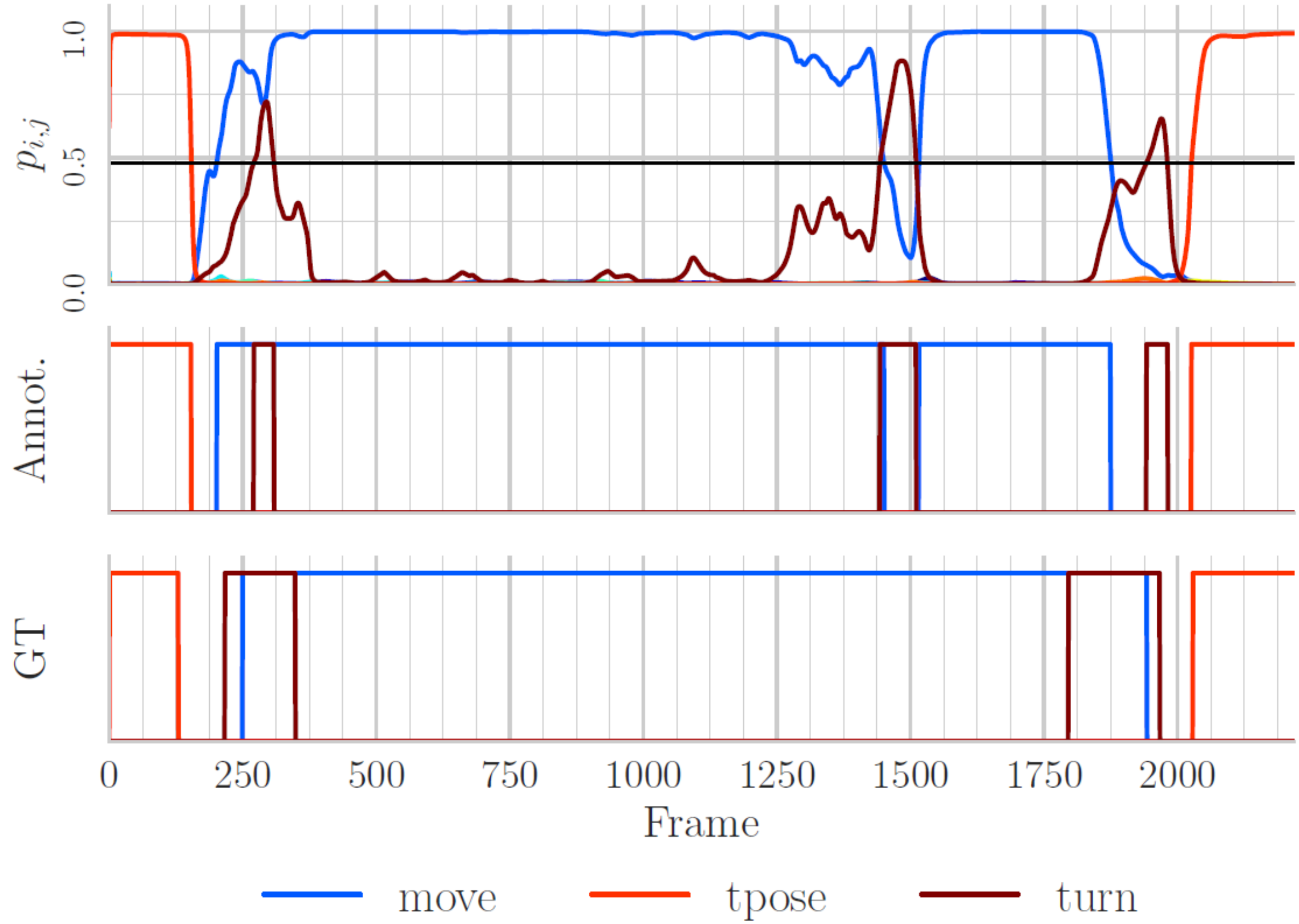
- Segmentation density impacts efficiency and effectiveness

LSTM-based semantic segmentation

- Learning a class assignment for each frame on training data
  - Sequences with their annotated parts are provided in advance
  - No similarity concept needed
- Online-LSTM model:
  - h_i – 1kD feature (1x1,024)
  - Sequence of n poses
  - m classes

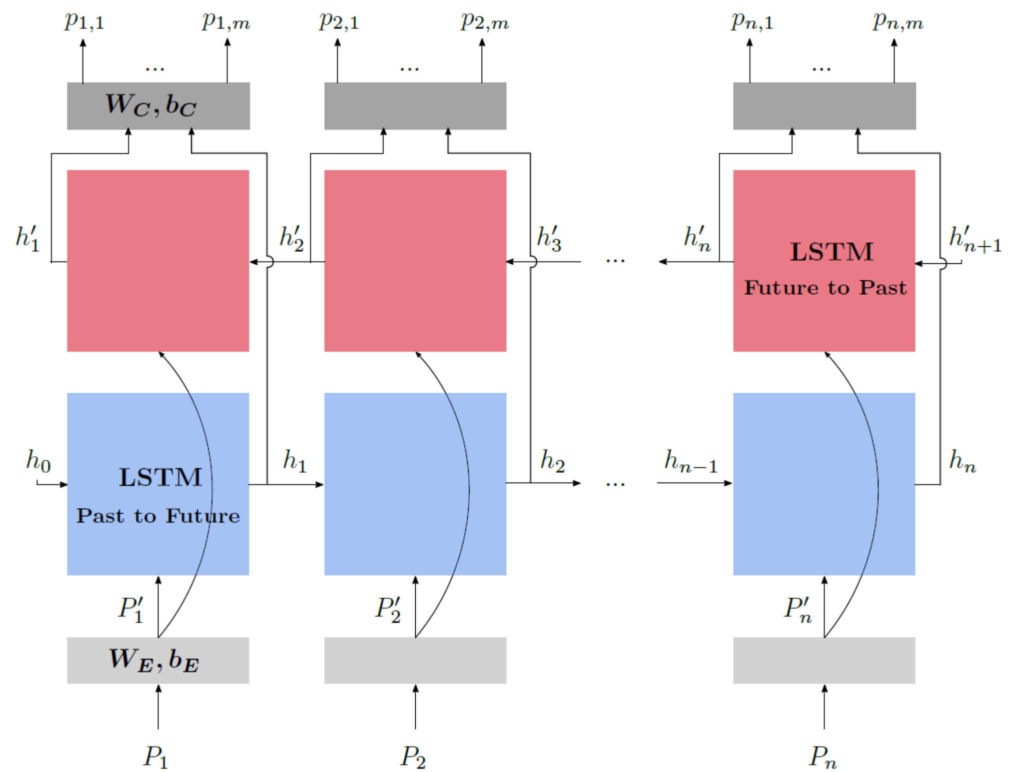


Output of Online-LSTM



Offline-LSTM model

- A bidirectional LSTM architecture to enhance the estimation of beginnings and endings of actions
- 1kD feature (2x512)
  - h'_i – 512D feature
  - h_i – 512D feature



HDM05 – long motions

- 102 long sequences ~ 68 minutes in total
- Ground truth – 1,464 short subsequences in 15 categories
  - Shortest and longest samples: 41 frames (0.3s) and 2,063 frames (17.2s)
  - Action classes corresponding to exercising activities:
    - Cartwheel
    - Exercise
    - Jump
    - Kick

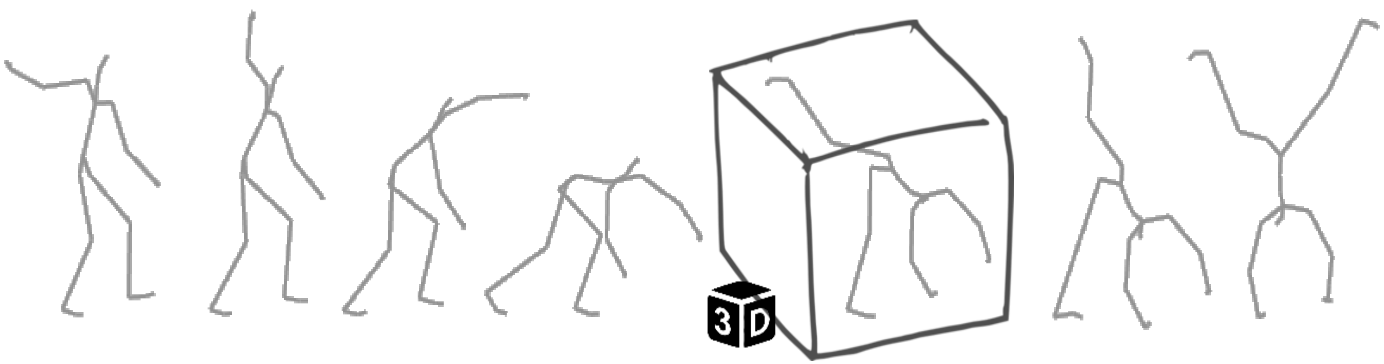
      ⋮
- Event detection scenario:
  - Actions in sequences of 17 mins used as representatives of classes
  - Sequences of 51mins used for online event detection

Accuracy measure

- F_1 score – a harmonic mean of recall and precision measured on the level of individual frames $$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$
  - Precision – the ratio of correctly annotated frames and all the algorithm-annotated frames
  - Recall – the ratio of correctly annotated frames and all the ground-truth annotated frames

	Training data	Test data	Training time	Per-frame efficiency			F_1 accuracy
				Extr.	Annot.	Total	
Muller et al. (2009)	24 min	60 min	N/A	1.9 ms	2.3 ms	4.2 ms	61.00 %
Muller + keyframes (2009)	24 min	60 min	N/A	1.9 ms	0.2 ms	2.1 ms	75.00 %
Segment-based ann. (2017)	17 min	51 min	2 h	7.1 ms	0.5 ms	7.6 ms	68.65 %
Online-LSTM (2018)	17 min	51 min	5 h	-	0.1 ms	0.1 ms	74.95 %
Offline-LSTM (2018)	17 min	51 min	3.5 h	-	0.1 ms	0.1 ms	78.78 %

7 Conclusions



Tutorial objectives:

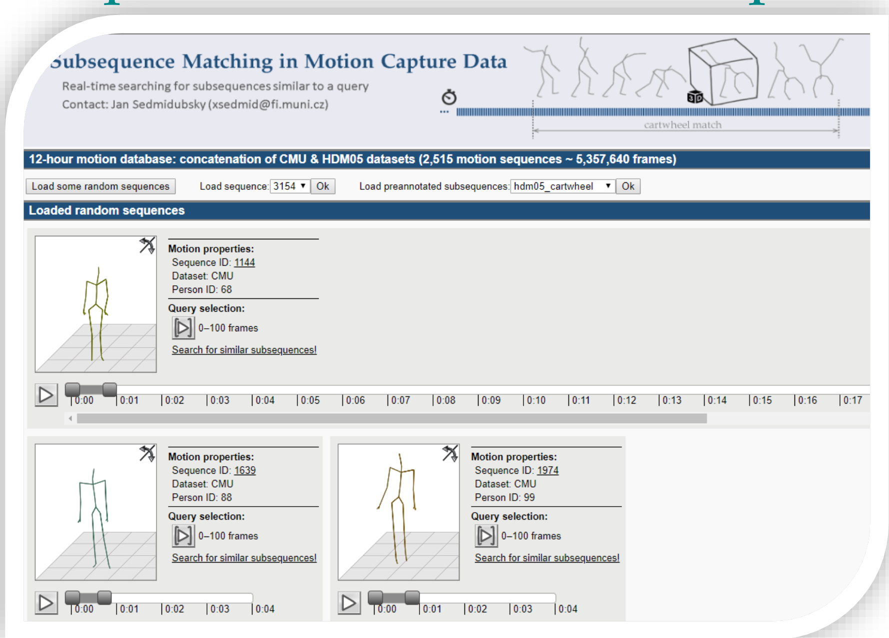
- To present challenges and existing principles for computerized processing of mocap capture data
  - **Presented operations** – similarity comparison, subsequence search, classification, semantic segmentation
- To focus not only on effectiveness but also on efficiency and exploit similarity search
- To apply modern machine-learning principles to automatically learn content-preserving movement features
- Presented approaches possibly applicable:
  - To any application field that processes motion data, e.g., medicine
  - To any spatio-temporal data ~ ground-reaction force (GRF) data

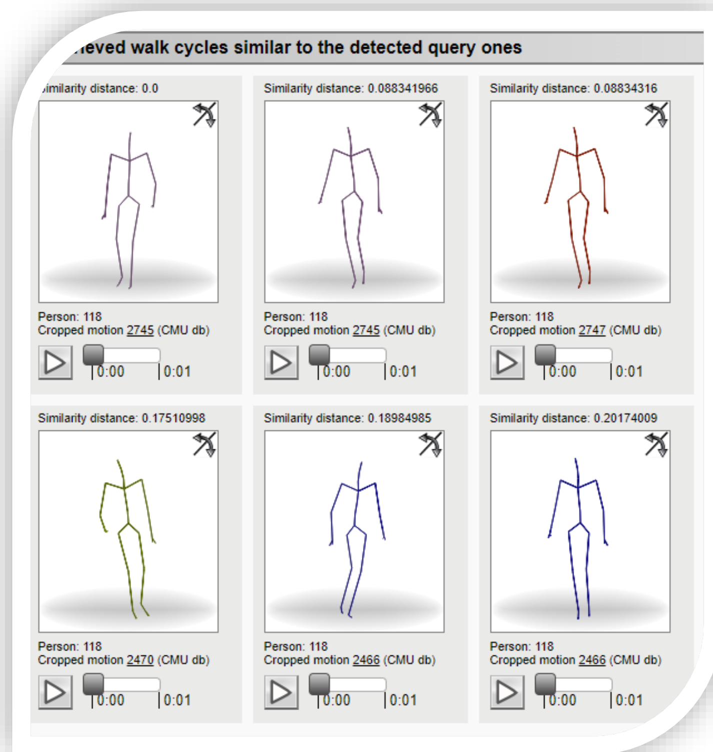
Classification/Subsequence search demo

- http://disa.fi.muni.cz/mocap-demo-classification/

Gait similarity search demo

- http://disa.fi.muni.cz/mmpi





Similarity Measures & Motion Features

- [Mathieu Barnachon, Saïda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. Pattern Recognition, 2014.]
- [Yong Du, Wei Wang, and Liang Wang. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. CVPR, 2015.]
- [Georgios Evangelidis, Gurkirt Singh, and Radu Horaud. Skeletal Quads: Human Action Recognition Using Joint Quadruples. ICPR, 2014.]
- [Harshad Kadu and C.-C. Jay Kuo. Automatic Human Mocap Data Classification. IEEE Transactions on Multimedia, 2014.]
- [Meinard Müller, Andreas Baak, and Hans-Peter Seidel. Efficient and Robust Annotation of Motion Capture Data. SCA, 2009.]
- [Jan Sedmidubsky, Petr Elias, and Pavel Zezula. Effective and Efficient Similarity Searching in Motion Capture Data. Multimedia Tools and Applications, 2018.]
- [Jan Sedmidubsky, Petr Elias, and Pavel Zezula. Enhancing Effectiveness of Descriptors for Searching and Recognition in Motion Capture Data, ISM 2017.]
- [Jan Sedmidubsky and Pavel Zezula. Probabilistic Classification of Skeleton Sequences. DEXA, 2018.]
- [Roshan Singh, Jagwinder Kaur Dhillon, Alok Kumar Singh Kushwaha, and Rajeev Srivastava. Depth based enlarged temporal dimension of 3D deep convolutional network for activity recognition. Multimedia Tools and Applications, 2018.]
- [Bin Sun, Dehui Kong, Shaofan Wang, Lichun Wang, Yuping Wang, and Baocai Yin. Effective human action recognition using global and local offsets of skeleton joints. Multimedia Tools and Applications, 2018.]
- [Chang Tang, Wanqing Li, Pichao Wang, and Lizhe Wang. Online human action recognition based on incremental learning of weighted covariance descriptors. Information Sciences, 2018.]
- [Yingying Wang and Michael Neff. Deep signatures for indexing and retrieval in large motion databases. Motion in Games, 2015.]

ACM MM 2018 Korea

Similarity Measures & Motion Features

- [D. Wu and L. Shao. Leveraging Hierarchical Parametric Networks for Skeletal Joints Based Action Segmentation and Recognition. CVPR, 2014.]
- [Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. Similarity Search: The Metric Space Approach. Advances in Database Systems, Vol. 32., Springer-Verlag. 220 pages.]
- [Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. Human Motion Analysis with Deep Metric Learning. ECCV, 2018.]

Similarity Searching

- [Zhigang Deng, Qin Gu, and Qing Li. Perceptually Consistent Examplebased Human Motion Retrieval. I3D, 2009.]
- [Y. Fang, K. Sugano, K. Oku, H. H. Huang, and K. Kawagoe. Searching human actions based on a multi-dimensional time series similarity calculation method. ICIS, 2015.]
- [Mubbasir Kapadia, I-kao Chiang, Tiju Thomas, Norman I Badler, and Joseph T Kider Jr. Efficient Motion Retrieval in Large Motion Databases. I3D, 2013.]
- [Björn Krüger, Anna Vögele, Tobias Willig, Angela Yao, Reinhard Klein, and Andreas Weber. Efficient Unsupervised Temporal Segmentation of Motion Data. IEEE Transactions on Multimedia, 2017.]
- [Jan Sedmidubsky, Petr Elias, and Pavel Zezula. Effective and Efficient Similarity Searching in Motion Capture Data. Multimedia Tools and Applications, 2018.]
- [Jan Sedmidubsky, Petr Elias, and Pavel Zezula. Searching for variable-speed motions in long sequences of motion capture data. Information Systems, 2018.]
- [Jan Sedmidubsky, Jakub Valcik, and Pavel Zezula. A Key-Pose Similarity Algorithm for Motion Data Retrieval. ACIVS, 2013.]
- [Jan Sedmidubsky, Pavel Zezula, and Jan Svec. Fast Subsequence Matching in Motion Capture Data. ADBIS, 2017.]
- [Pavel Zezula. Similarity Searching for the Big Data. Mob. Netw. Appl., 2015.]
- [Pavel Zezula. Similarity Searching for Database Applications. ADBIS, 2016.]
- [Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. Similarity Search: The Metric Space Approach. Advances in Database Systems, Vol. 32., Springer-Verlag. 220 pages.]

Classification

- [Fabien Baradel, Christian Wolf, and Julien Mille. Human Action Recognition: Pose-based Attention draws focus to Hands. ICCV Workshop on Hands in Action, 2017.]
- [Mathieu Barnachon, Saïda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. Pattern Recognition, 2014.]
- [Judith Butepage, Michael J. Black, Danica Kragic, and Hedvig Kjellstrom. Deep Representation Learning for Human Motion Prediction and Classification. CVPR, 2017.]
- [Yong Du, Wei Wang, and Liang Wang. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. CVPR, 2015.]
- [Georgios Evangelidis, Gurkirt Singh, and Radu Horaud. Skeletal Quads: Human Action Recognition Using Joint Quadruples. ICPR, 2014.]
- [Harshad Kadu and C.-C. Jay Kuo. Automatic Human Mocap Data Classification. IEEE Transactions on Multimedia, 2014.]
- [Sohaib Laraba, Mohammed Brahimi, Joelle Tilmanne, and Thierry Dutoit. 3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images. Computer Animation and Virtual Worlds, 2017.]
- [Chaolong Li, Zhen Cui, Wenming Zheng, Chunyan Xu, and Jian Yang. Spatio-Temporal Graph Convolution for Skeleton Based Action Recognition. AAAI, 2018.]
- [Jun Liu, Amir Shahroudy, Dong Xu, and GangWang. Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. ECCV, 2016.]
- [Jun Liu, Gang Wang, Ling-Yu Duan, Ping Hu, and Alex C. Kot. Skeleton Based Human Action Recognition with Global Context-Aware Attention LSTM Networks. IEEE Transactions on Image Processing, 2018.]
- [Juan C. Nunez, Raul Cabido, Juan J. Pantrigo, Antonio S. Montemayor, and Jose F. Velez. Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity recognition and hand gesture recognition. Pattern Recognition, 2018.]

Classification

- [Jan Sedmidubsky and Pavel Zezula. Probabilistic Classification of Skeleton Sequences. DEXA, 2018.]
- [Roshan Singh, Jagwinder Kaur Dhillon, Alok Kumar Singh Kushwaha, and Rajeev Srivastava. Depth based enlarged temporal dimension of 3D deep convolutional network for activity recognition. Multimedia Tools and Applications, 2018.]
- [Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. CoRR abs/1611.06067, 2016.]
- [Bin Sun, Dehui Kong, Shaofan Wang, Lichun Wang, Yuping Wang, and Baocai Yin. Effective human action recognition using global and local offsets of skeleton joints. Multimedia Tools and Applications, 2018.]
- [Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks. AAAI, 2016.]
- [Pattreeya Tanisaro and Gunther Heidemann. An Empirical Study on Bidirectional Recurrent Neural Networks for Human Motion Recognition. TIME, 2018.]

Semantic Segmentation

- [Said Yacine Boulahia, Eric Anquetil, Franck Multon, and Richard Kulpa. CuDi3D: Curvilinear displacement based approach for online 3D action detection. Computer Vision and Image Understanding, 2018.]
- [Judith Butepage, Michael J. Black, Danica Kragic, and Hedvig Kjellstrom. Deep Representation Learning for Human Motion Prediction and Classification. CVPR, 2017.]
- [Petr Elias, Jan Sedmidubsky, and Pavel Zezula. A Real-Time Annotation of Motion Data Streams. ISM, 2017.]
- [Sheng Li, Kang Li, and Yun Fu. Early Recognition of 3D Human Actions. ACM Trans. Multimedia Comput. Commun. Appl., 2018.]
- [Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning Activity Progression in LSTMs for Activity Detection and Early Detection. CVPR, 2016.]
- [Meinard Müller, Andreas Baak, and Hans-Peter Seidel. Efficient and Robust Annotation of Motion Capture Data. SCA, 2009.]
- [Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection. IEEE Transactions on Image Processing, 2018.]
- [Chang Tang, Wanqing Li, Pichao Wang, and Lizhe Wang. Online human action recognition based on incremental learning of weighted covariance descriptors. Information Sciences, 2018.]
- [D. Wu and L. Shao. Leveraging Hierarchical Parametric Networks for Skeletal Joints Based Action Segmentation and Recognition. CVPR, 2014.]
- [Yan Xu, Zhengyang Shen, Xin Zhang, Yifan Gao, Shujian Deng, YipeiWang, Yubo Fan, and EricI-Chao Chang. Learning multi-level features for sensor-based human action recognition. Pervasive and Mobile Computing, 2017.]
- [Xin Zhao, Xue Li, Chaoyi Pang, Quan Z. Sheng, Sen Wang, and Mao Ye. Structured Streaming Skeleton – A New Feature for Online Human Gesture Recognition. ACM Trans. Multimedia Comput. Commun. Appl., 2014.]

Resources

ACM MM 2018 Korea

Presentations

- [Lukas Masuch: Deep Learning – The Past, Present and Future of Artificial Intelligence, 2015]

Funding

- Supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822)