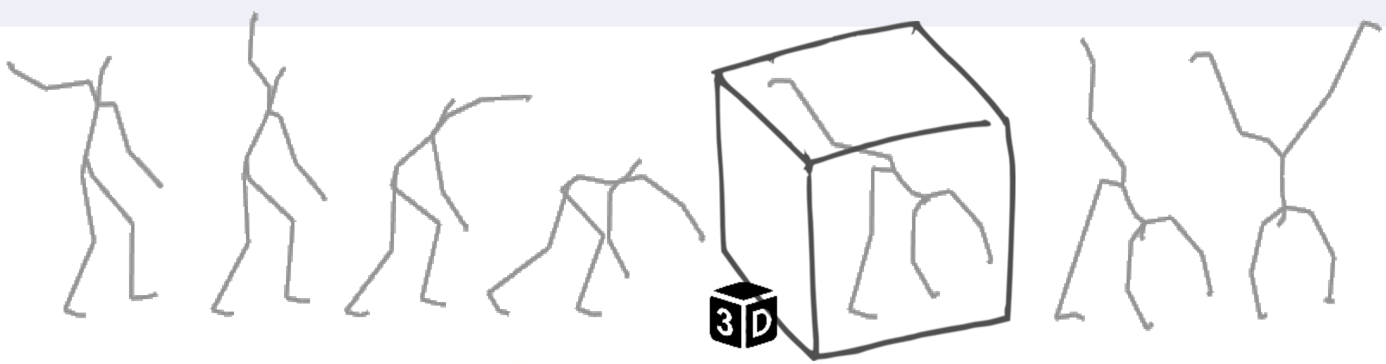
ACM ICMR 2019 Ottawa

Similarity Search in 3D Human Motion Data

Jan Sedmidubsky
xsedmid@fi.muni.cz

Pavel Zezula
zezula@fi.muni.cz

[Jan Sedmidubsky and Pavel Zezula. Similarity Search in 3D Human Motion Data. ACM International Conference on Multimedia Retrieval (ICMR). ACM, pp. 5–6, 2019.]
https://dl.acm.org/citation.cfm?id=3326589



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

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

Supported by the Czech Science Foundation project "Searching, Mining, and Annotating Human Motion Streams" No. GA19-02033S.

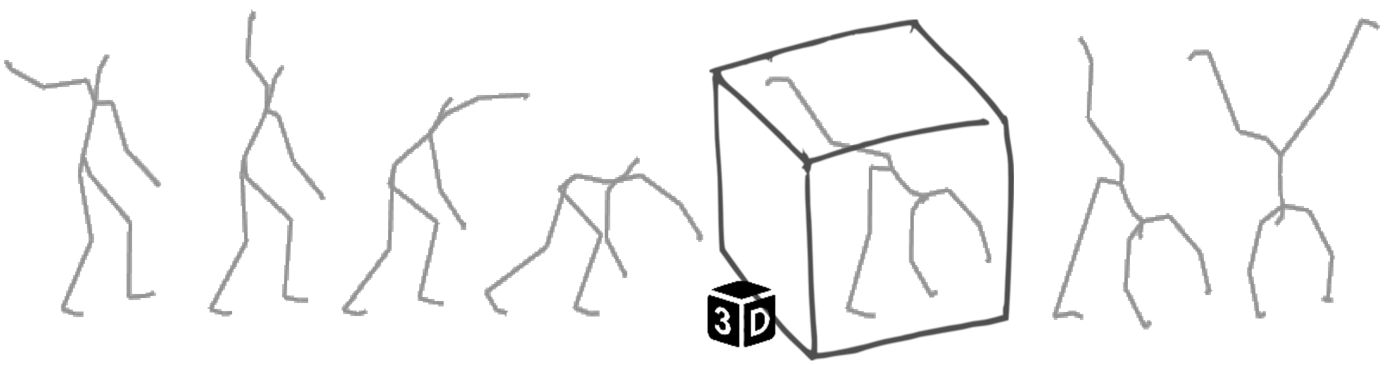
Outline

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

- - - - - - - - - Coffee break - - - - - - - - -

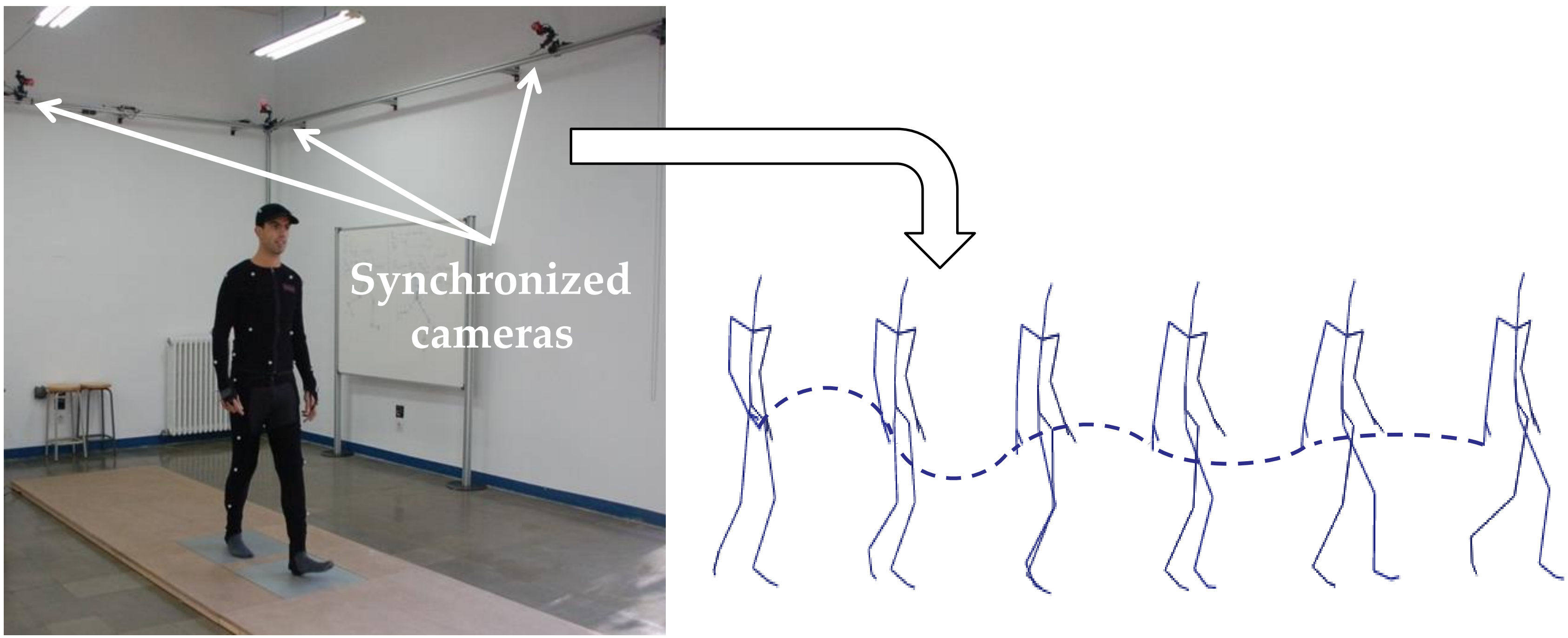
5) Metric Searching as a Data-Access Paradigm
6) Action Recognition
7) Indexing and Searching in Long Motion Sequences
   - Subsequence Search in Long Sequences
   - Stream-based Event Detection
8) Conclusions and Discussion

1 Motion Capture Data: Acquisition and Applications



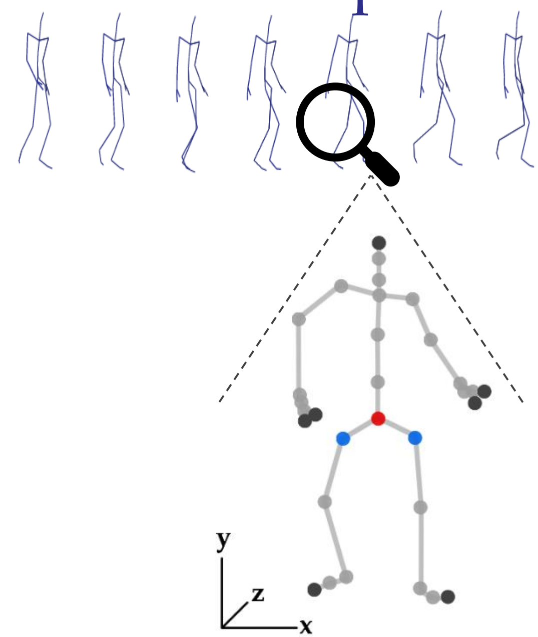
Motion Capture Data ~ MoCap Data ~ Motion Data

- 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 in a given time moment
    - 3D positions of body landmarks, denoted as joints

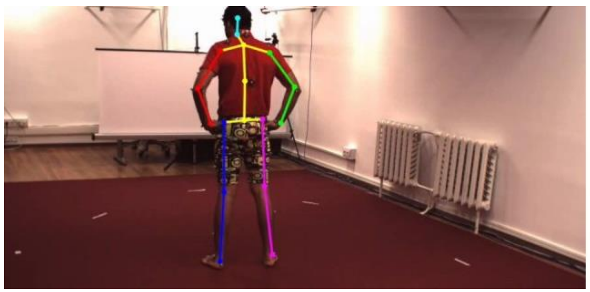


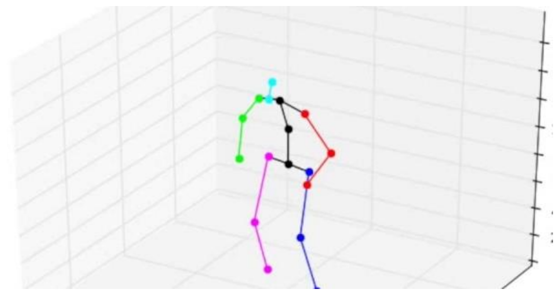
Pose captured in a given time moment

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

3D motion capture principles

- Estimating 3D poses from a single 2D video camera
  - Not so precise but a great applicability





- Applying 3D motion capture technologies
  - Precise but expensive and with a limited applicability
  - Technologies:
    - Optical
      - Marker-based (invasive)
      - Marker-less (non-invasive)
    - Other – inertial, magnetic, mechanical, radio frequency

Accuracy of 3D motion capture 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

1.2 3D Motion Capture Devices

ACM ICMR 2019 Ottawa

3D motion capture devices

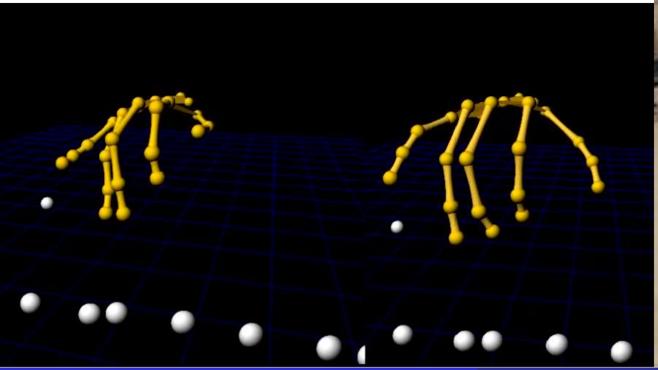
- Optical-based devices are the most commonly used
- Advantages/disadvantages:
  - Invasive – accurate | large space | markers | expensive
    - Vicon, MotionAnalysis
  - Non-invasive – no markers | small space
    - Accurate but expensive – Organic Motion
    - Less accurate but cheap – Microsoft Kinect, 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**

ACM ICMR 2019 Ottawa

Applications

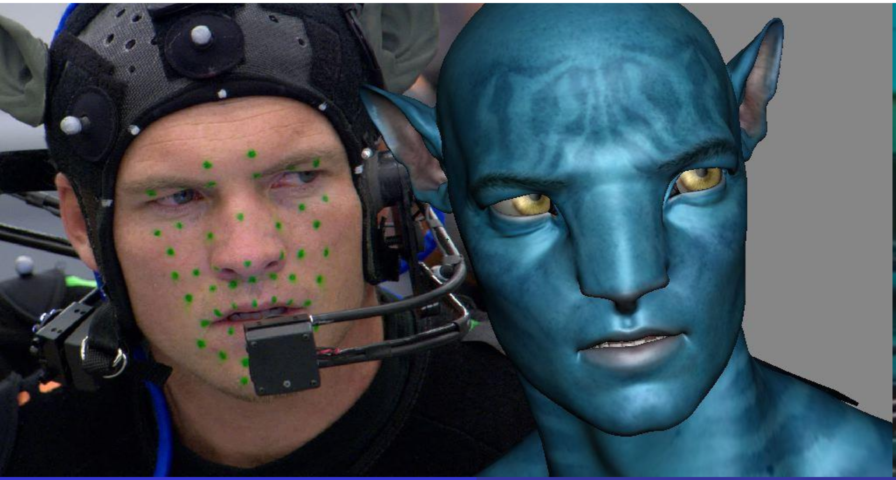
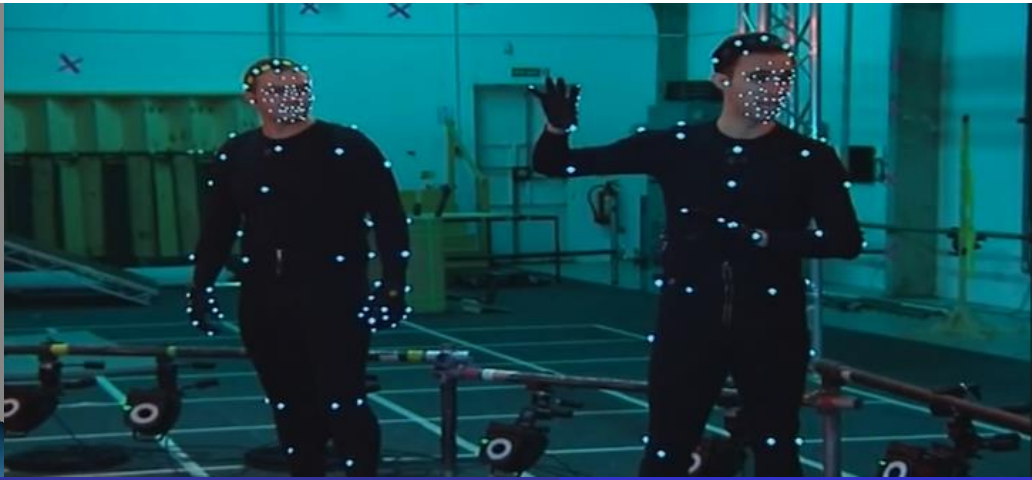
- Many application domains where motion data have a great potential to be utilized and automatically processed
  - Computer animation & human-computer interaction
  - Military
  - Sports
  - Medicine
  - Other domains



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
- Generate artificial motions by merging real movements that follow each other

1.3 Applications

Human computer interaction, augmented reality

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

1.3 Applications

Sports

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

1.3 Applications

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

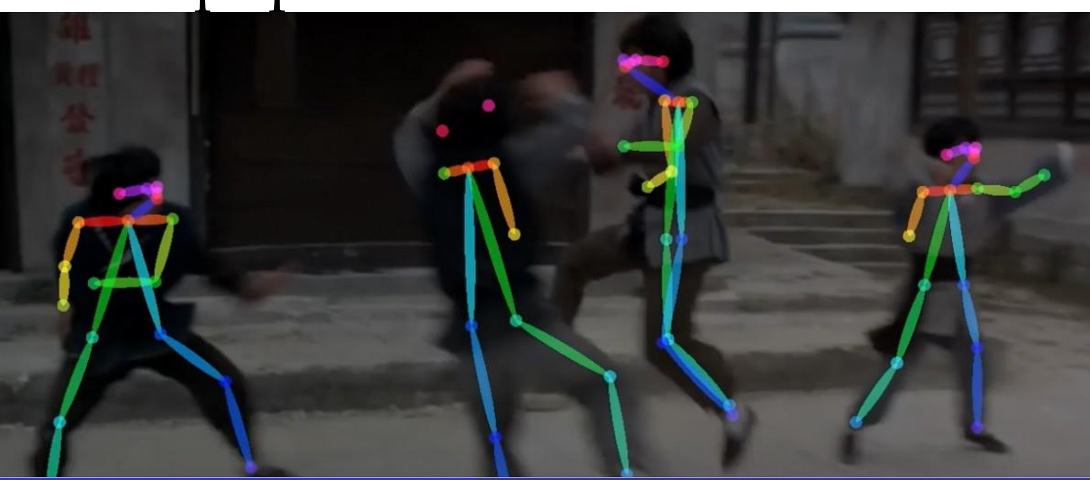
ACM ICMR 2019 Ottawa

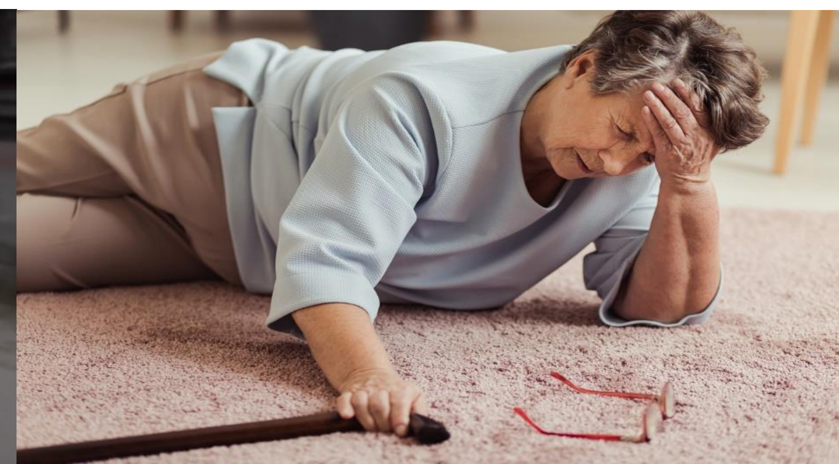
Military

- 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

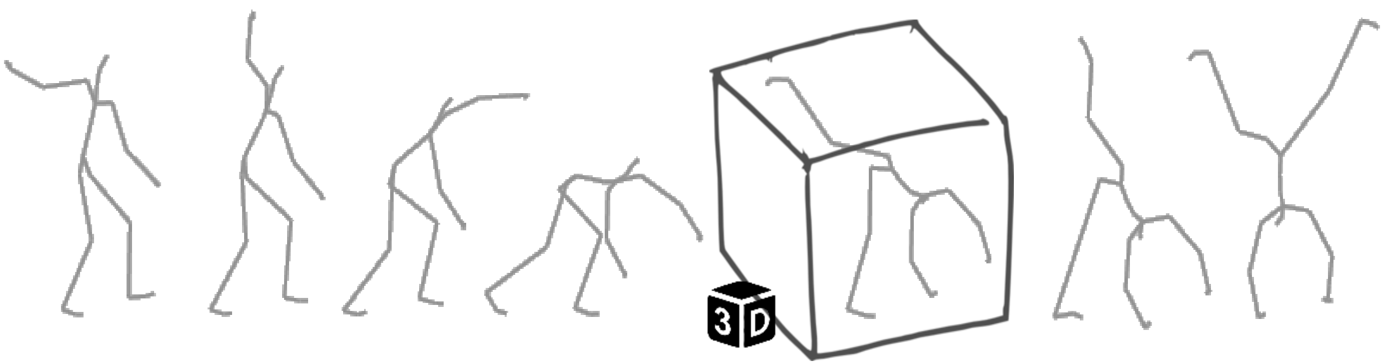
Other domains

- Security – identification of persons based on their style of walking (~ gait recognition)
- Smart-homes – detection of falls of elderly people
- Construction-sites – identification of unsafe acts, e.g., speed limit violations of equipment or close distance between equipment and workers





2 Challenges in Motion Data Processing



2 The Big Data Corollaries

ACM ICMR 2019 Ottawa

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

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

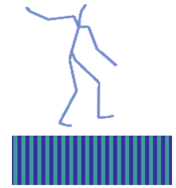
Motion data specifics

- 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

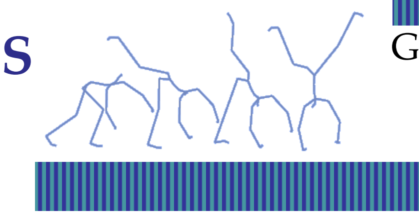
2.1 Data – Types of Motions

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)

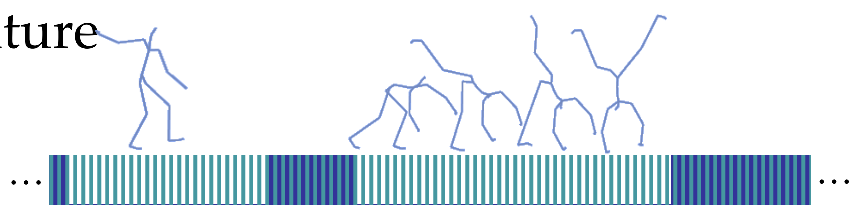
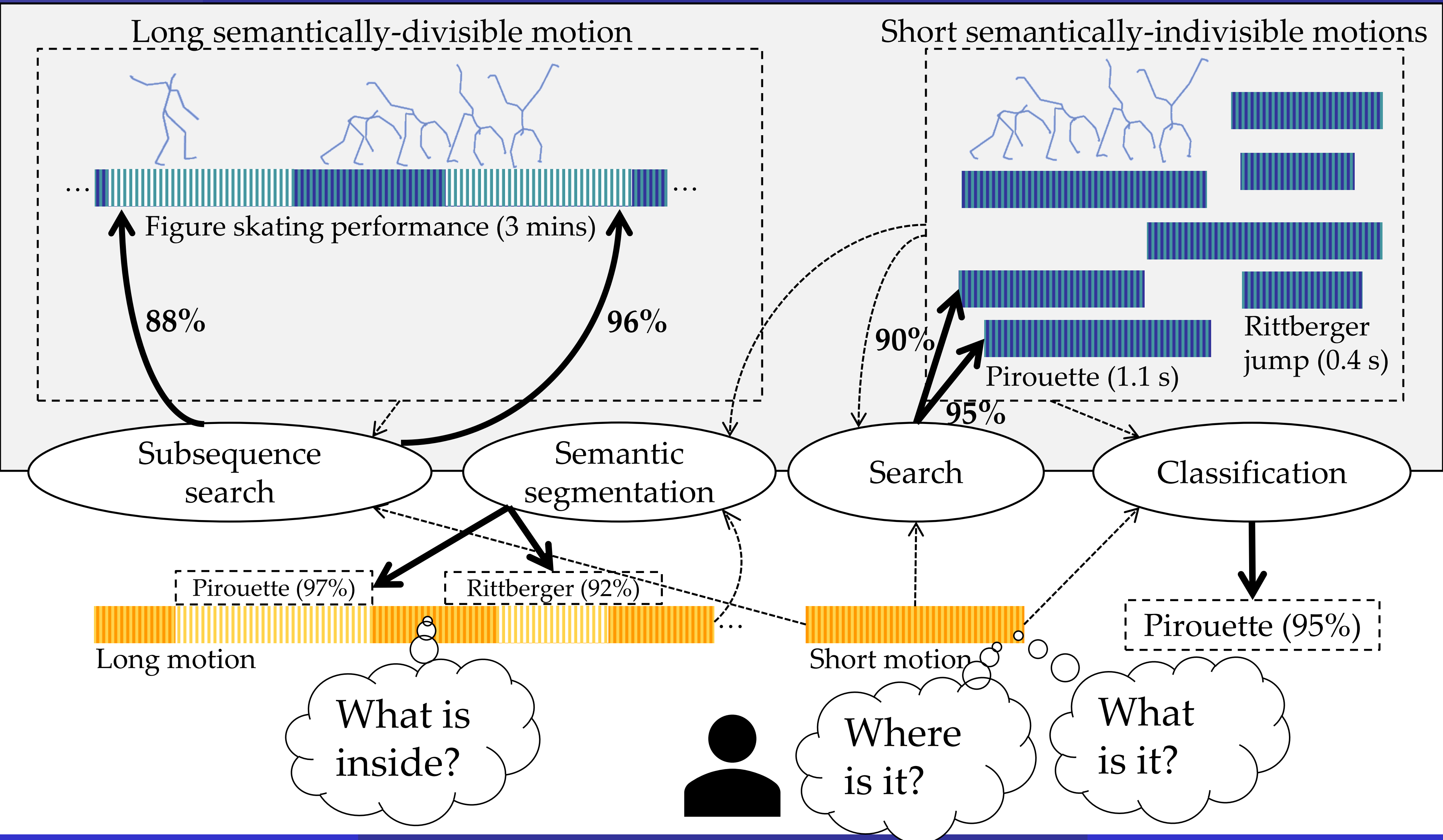


Figure skating performance (3 mins)

2.3 Motion-Analysis Operations

ACM ICMR 2019 Ottawa



Motion-analysis operations

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

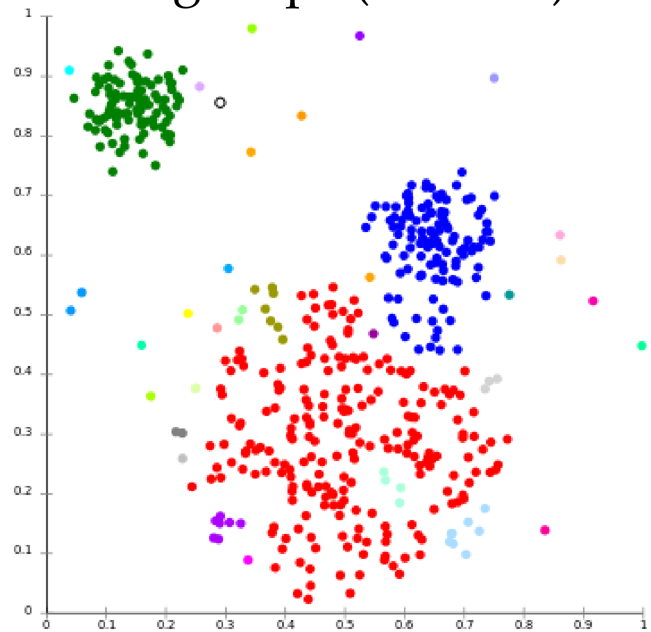
  ⋮

2.3 Other Operations – Clustering

ACM ICMR 2019 Ottawa

Clustering

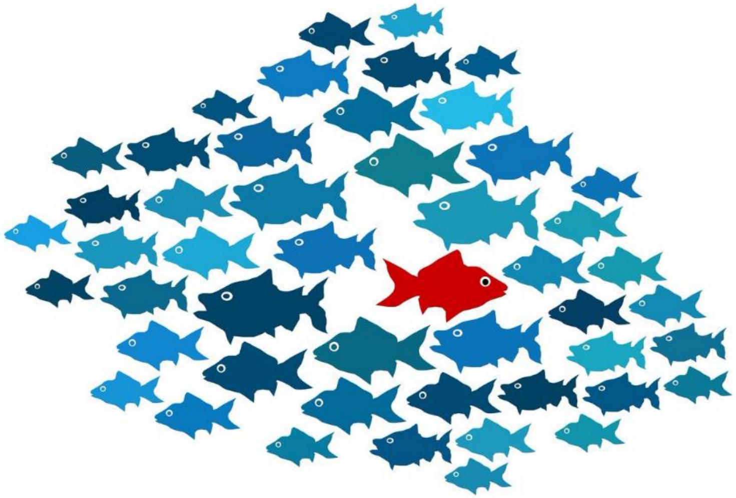
- Suppose each motion as a point in *n*-dimensional space
- Grouping motions in action collections
  - Motions in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)
- Useful for statistical data analysis

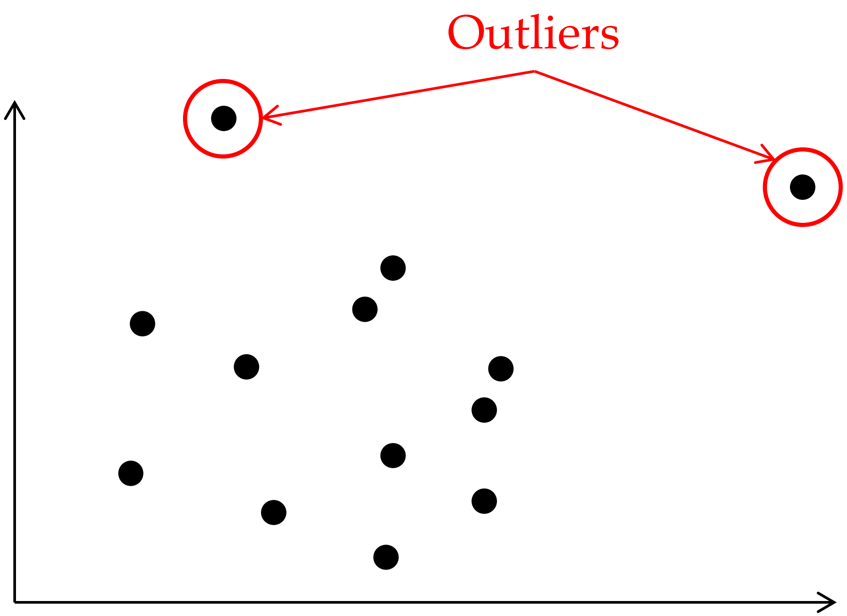


ACM ICMR 2019 Ottawa

Outlier detection

- Identifying motions which significantly deviate from other motion entities

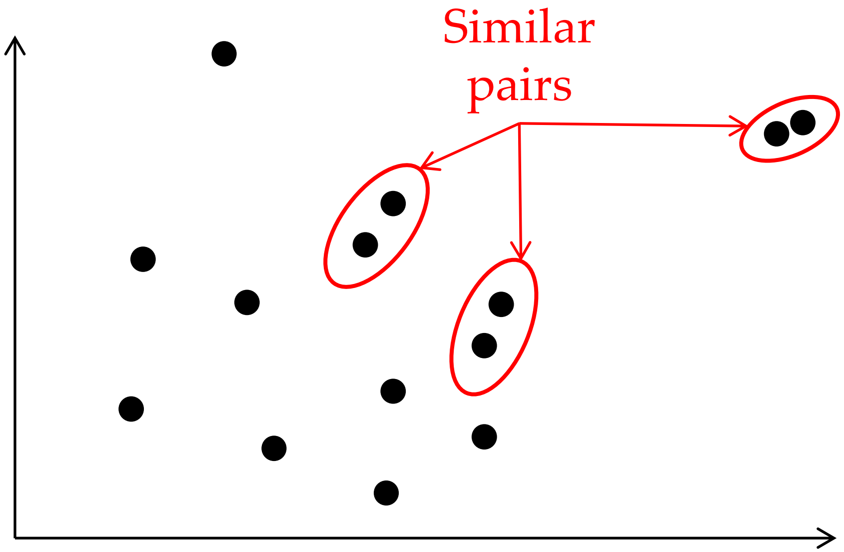




ACM ICMR 2019 Ottawa

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



Summary of operations

OPERATION	OPERATION DATA (KNOWLEDGE BASE)	USER INPUT (QUERY)	OPERATION RESULT
Search	Actions	Action	Actions similar to the query action
Subsequence search	Long motions	Action	Positions of query-similar subsequences
Classification	Labelled (categorized) actions	Action	Class of the examined action
Semantic segmentation	Labelled (categorized) actions	Long motion	Positions of detected 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

ACM ICMR 2019 Ottawa

Quotations from the social psychology literature

- 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 and context-dependent

Are they similar?

Are they similar?

ACM ICMR 2019 Ottawa

Are they similar?

ACM ICMR 2019 Ottawa

Are they similar?

ACM ICMR 2019 Ottawa

Prototypicality or centrality

- Not symmetric

3.1 Real-Life Similarity

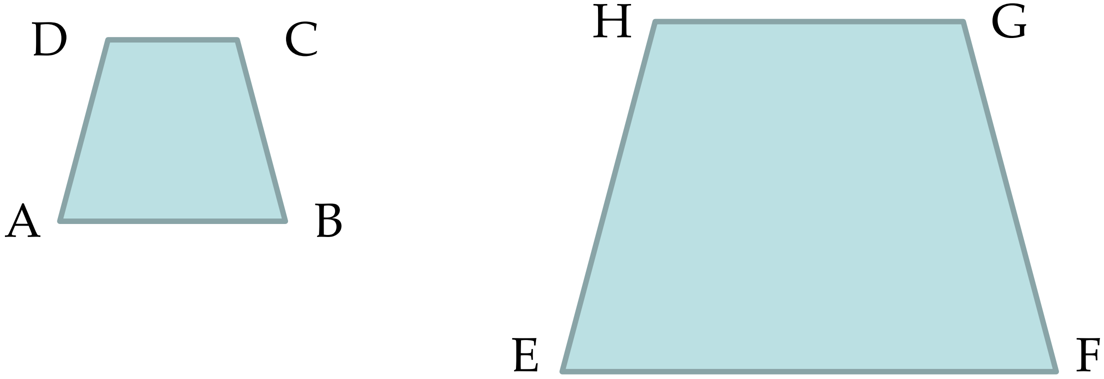
Context/Data/Environment dependent

- Circumstances alter similarities

ACM ICMR 2019 Ottawa

We learned from school

- Two polygons are similar to each other, if:
  1) Their corresponding angles are congruent
     - $\angle A = \angle E; \angle B = \angle F; \angle C = \angle G; \angle D = \angle H$, 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

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

Majority of current data is **unstructured**,

possibly only structured on display

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*

3.1 Iterative and Interactive Nature of Contemporary Searching

- When we search, our next actions are reactions to the stimuli of previous search results
- What we find is changing what we seek
- In any case, search must be:

fast, *simple*, and *relevant*

3.2 Metric Space: A Geometric 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
  - $\forall\, x, y, z \in \mathcal{D}$:

$d(x, y) > 0$ – *non-negativity*

$d(x, y) = 0 \Leftrightarrow x = y$ – *identity*

$d(x, y) = d(y, x)$ – *symmetry*

$d(x, y) \leq d(x, z) + d(z, y)$ – *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

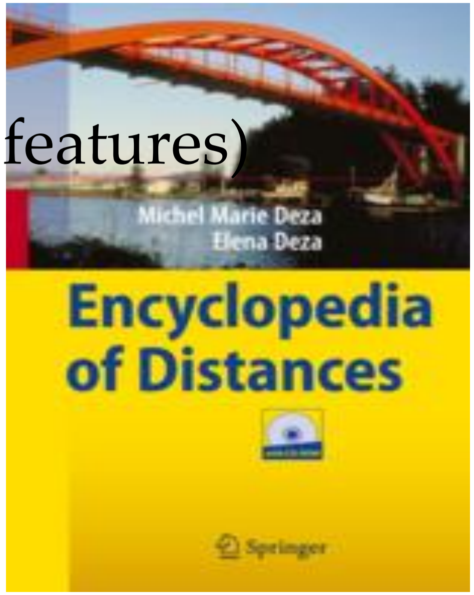
- Jaccard’s coefficient – for sets A, B

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

3.2 Metric Space – Distance Functions

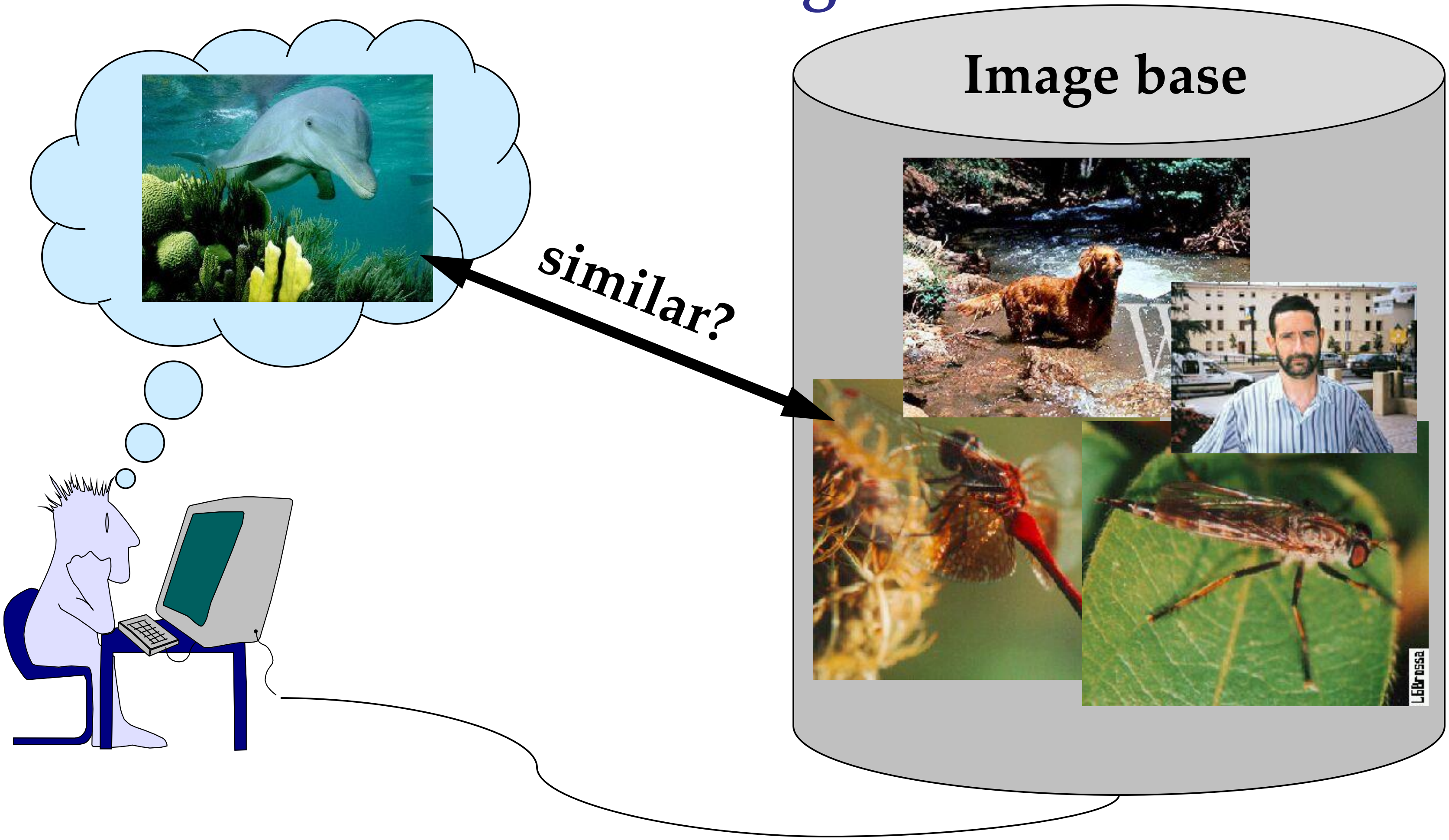
Example of other distance functions

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



ACM ICMR 2019 Ottawa

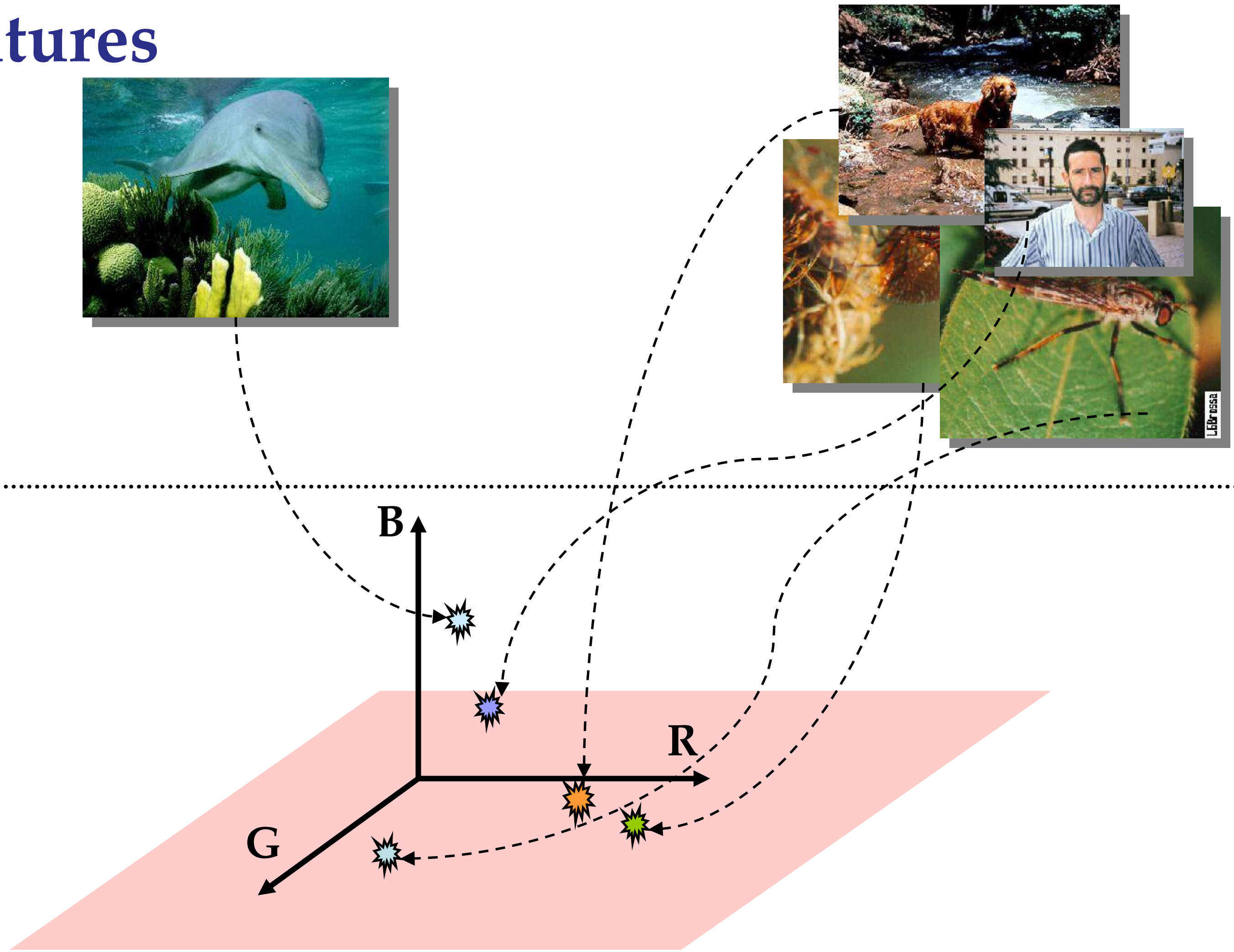
Content-based search in images



ACM ICMR 2019 Ottawa

Extracting features

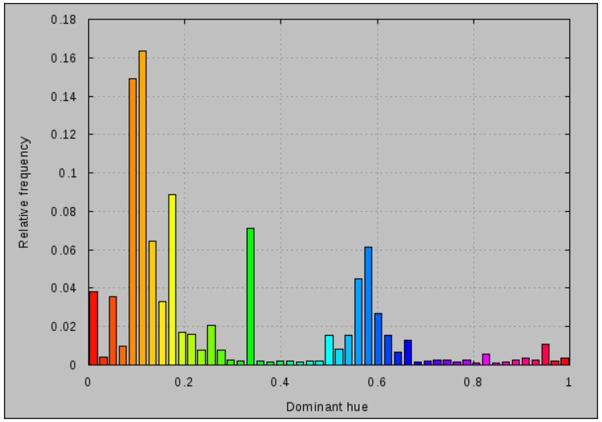
Image level



Feature level

Examples of features – MPEG-7

- MPEG-7 multimedia content descriptor standard ~ 2000
  - Global feature descriptors – color, shape, texture, etc.
  - One high-dimensional vector per image and feature
  - Minkovski distance used

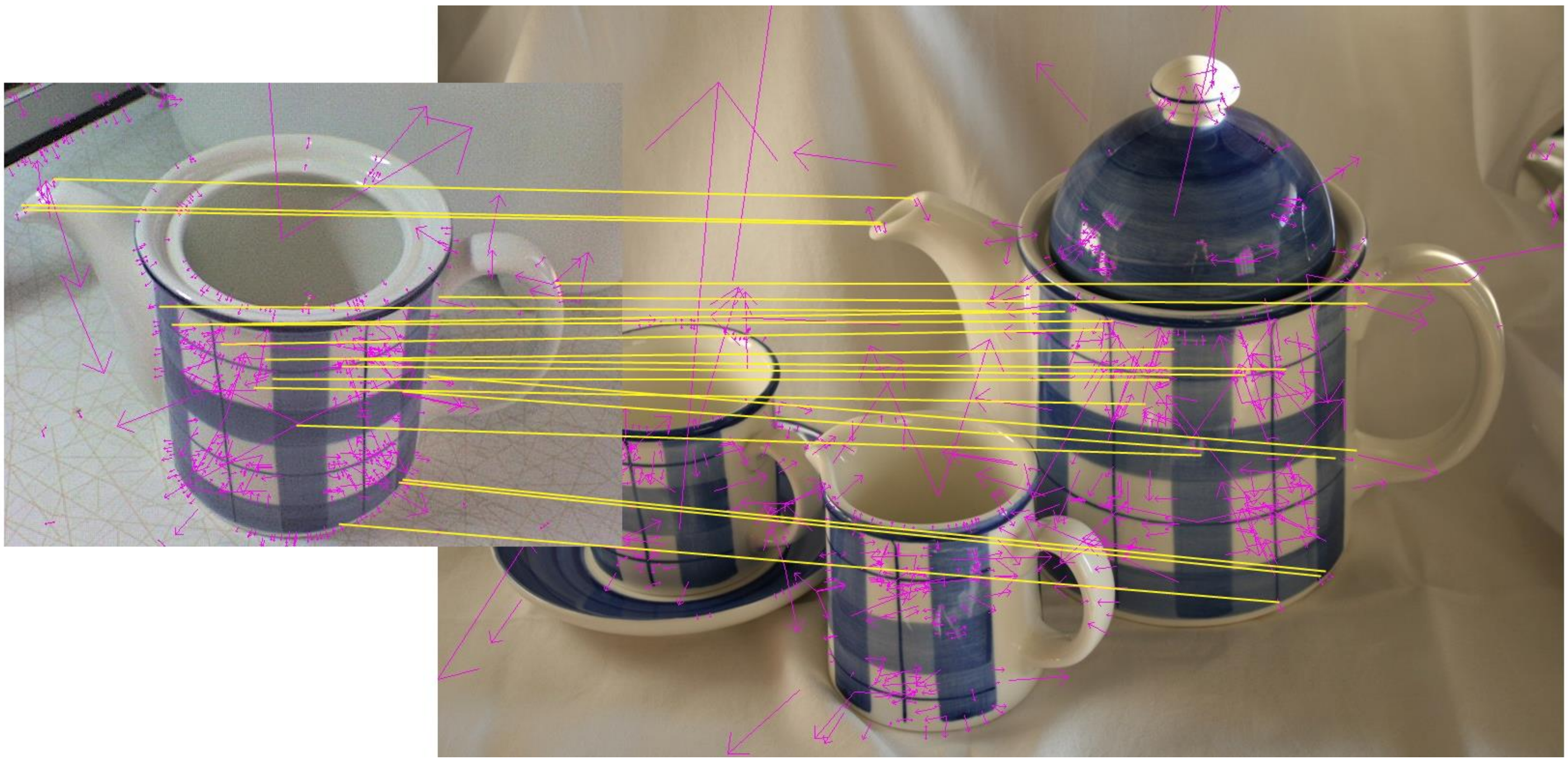


Examples of features

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

ACM ICMR 2019 Ottawa

Finding correspondence



3.2 Visual Similarity – Biometrics

ACM ICMR 2019 Ottawa

Fingerprints

- Minutiae detection:
  - Detect ridges (endings and branching)
  - Represented as a sequence of minutiae
    - P=($(r_1, e_1, \theta_1), \ldots, (r_m, e_m, \theta_m)$)
    - 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

3.2 Visual Similarity – Biometrics

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

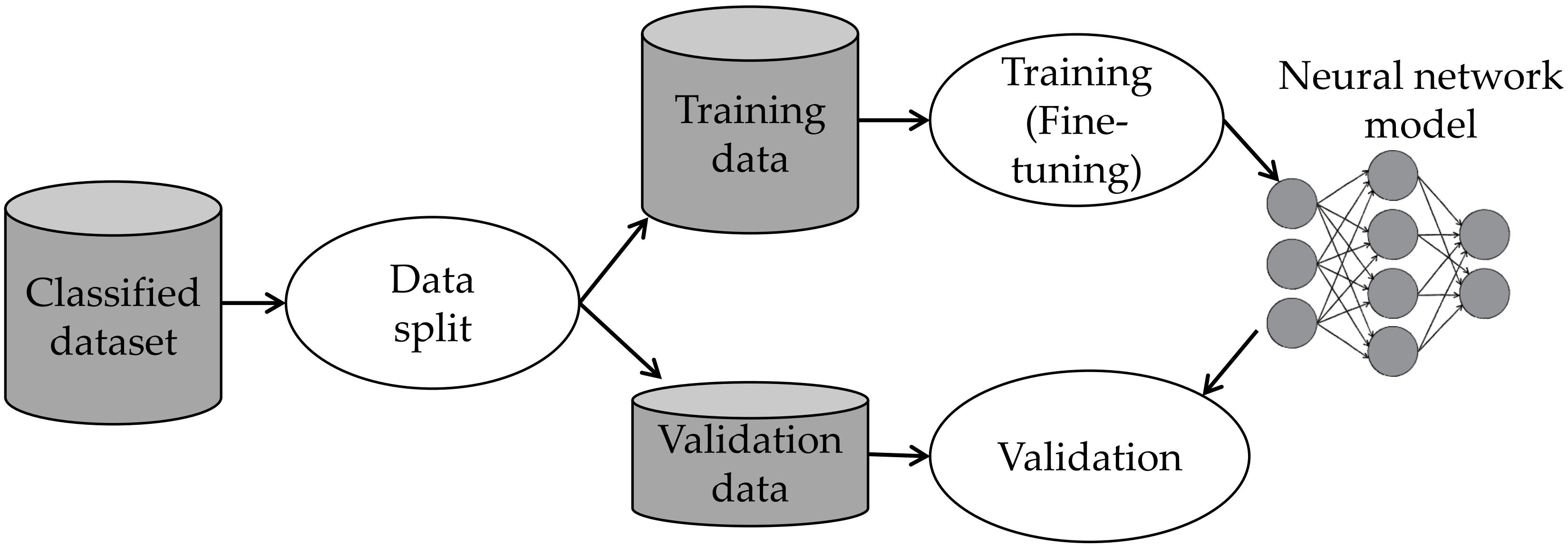
$$H(F,G) = \max(h(F,G), h(G,F))$$

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

Multiple visual aspects

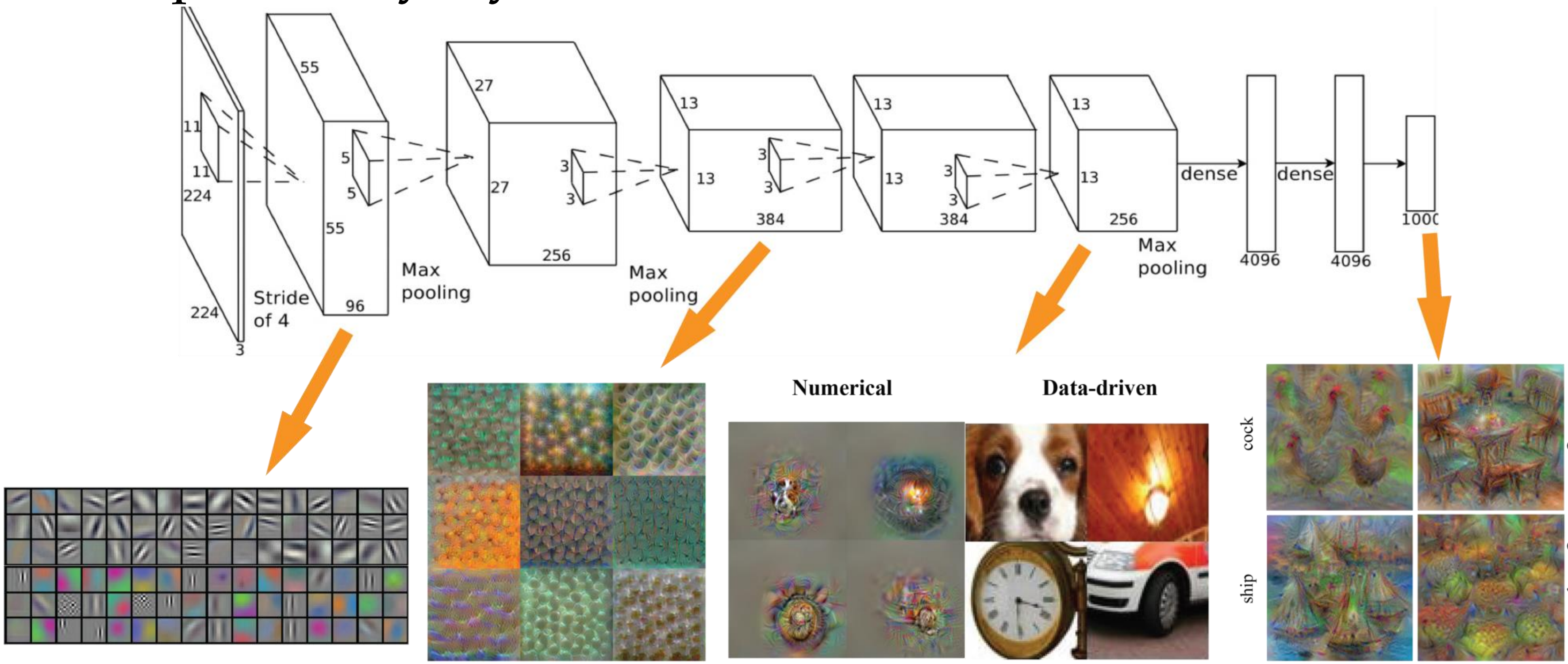
Current approaches in feature extraction

- Technology of neural networks
  - Convolutional neural networks (CNN)
  - Recurrent neural networks (RNN)



Convolutional neural network (CNN) – AlexNet

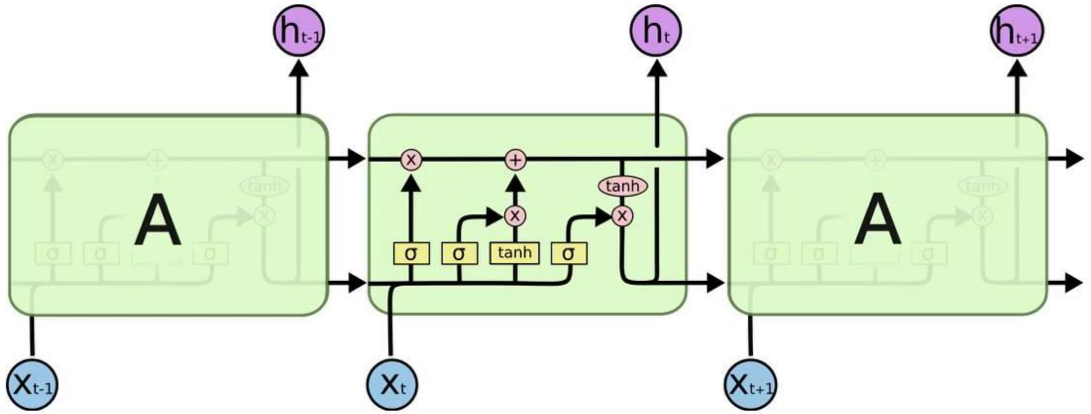
- The last layer with 1,000 output categories
- Output of any layer can be used as a feature



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

Recurrent neural networks (RNN)

- 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

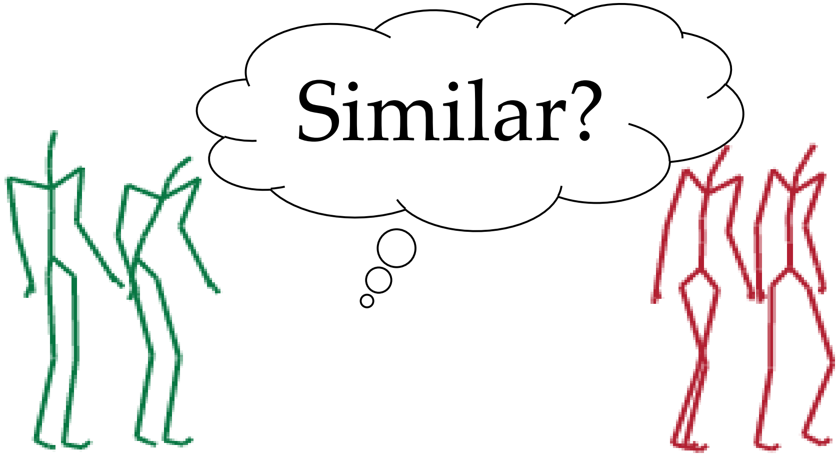
- It is no magic! Just statistics in a black box, but exceptional effective at learning patterns
- Powerful computational infrastructure can be applied, e.g., GPU cards

- Deep learning can be used not only for classification but is also able to provide content preserving feature vectors
- When calibrated by an L_p distance, good quality similarity estimates can be obtained

Similarity search demos – extensibility

- 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-ui/
- Fingerprints: http://disa.fi.muni.cz/fingerprints/
- Time series: http://disa.fi.muni.cz/subseq/

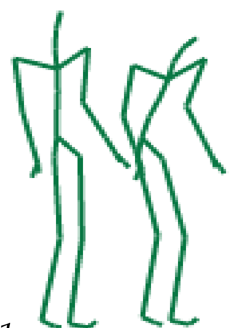
4 Similarity of Actions



Similarity of short motion sequences (~ actions)

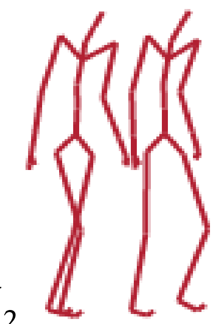
- Determining similarity is needed everywhere, e.g., for clustering, classification, searching, semantic segmentation
  - Similarity measure = features + distance function

M_1



How similar are the motions?

M_2



Feature extraction process

<0, 0, 5.2, 8.1, 0, 2.3, -1.1, 0, …>

F_{M_1}

<0, 0.1, 6.2, 7.1, 0, 2.9, -2.1, 0, …>

F_{M_2}

$$dist\left(F_{M_1}, F_{M_2}\right) = 8.56$$

4.1 Challenges of Similarity Measures

Objective

- To propose an effective and efficient similarity measure, i.e., content-preserving features + fast distance function

Problems

- 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)*
- 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 design

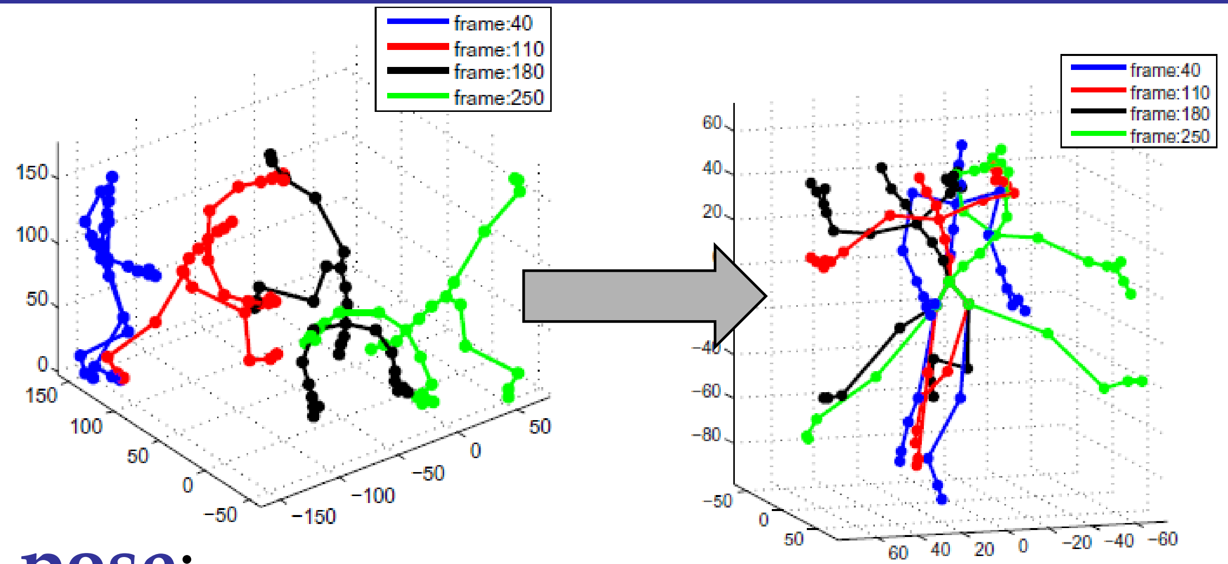
- Data normalization – optional pre-processing step
  - Space-dependent/invariant motion data
  - Skeleton-size variability
- Granularity of features
  - Pose features – a set of times series + alignment function
    - E.g., joint angle rotations, distances between joints
  - Motion features – a fixed-length vector + L_p metric
    - E.g., average velocity of individual joints, lengths of joint trajectories
- Engineering
  - Hand-crafted features – manual feature engineering
  - Machine-learned features – learning features automatically

4.2 Input Data Normalization

ACM ICMR 2019 Ottawa

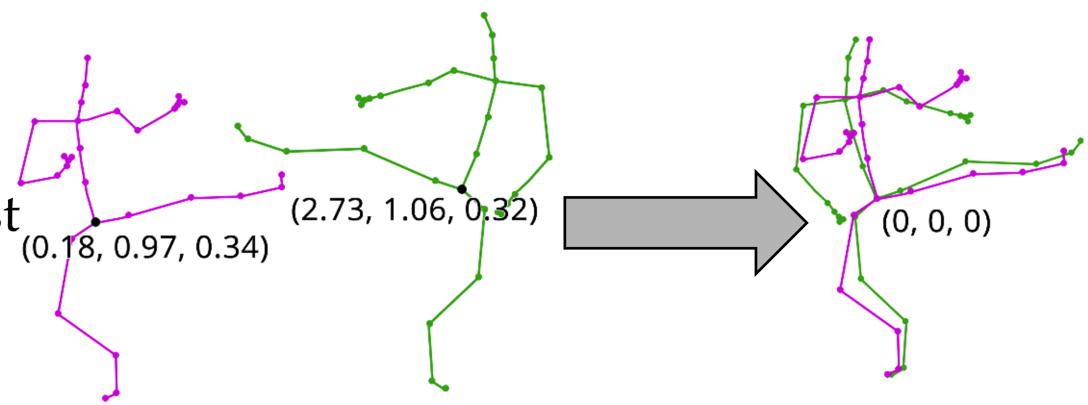
Normalization of:

- Position
- Orientation
- Skeleton size



Normalizing each pose:

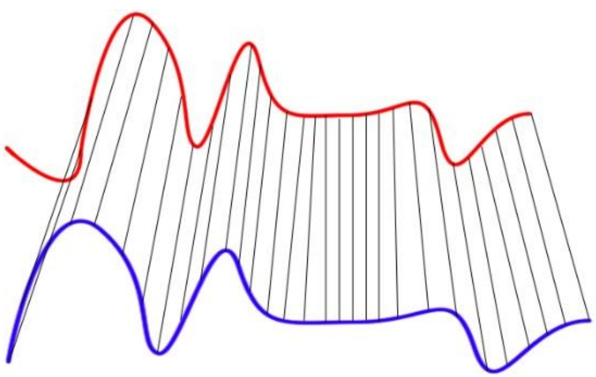
- Independently
- Conditionally
  - E.g., relatively to the first normalized pose



ACM ICMR 2019 Ottawa

Hand-crafted features

- Very good knowledge of data domain is needed
- Very specialized in what they express
- Lower descriptive power compared to ML approaches
- Usually extracted for individual poses ~ time series
  - E.g., time series of joint angle rotations + Dynamic Time Warping (DTW)



Dynamic Time Warping Matching

Existing hand-crafted-based approaches

- 17D motion feature – 17 scalars describing gait velocity, stride length, step frequency, etc.

  [Pradhan et al., Automated classification of neurological disorders of gait using spatio-temporal gait parameters, Journal of Electromyography and Kinesiology, 2015]

  [classification of neurological disorders of gait]

- 6D pose features – pair-wise distances between joints

  [Sedmidubsky et al., Gait Recognition Based on Normalized Walk Cycles, ISVC 2013]

  [gait recognition]

- 28D pose features – 28 joint-angle rotations

  [Sedmidubsky et al., A key-pose similarity algorithm for motion data retrieval, ACIVS 2013]

  [action searching]

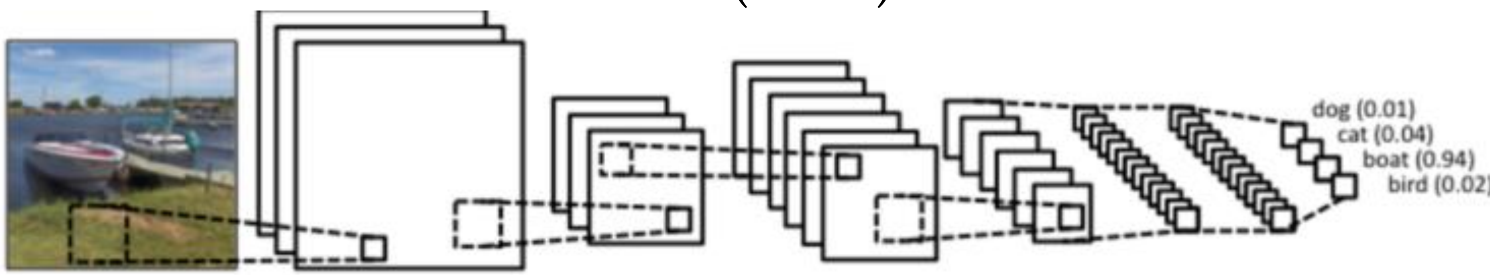
- 40D pose features – 40 relational frame-based characteristics

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

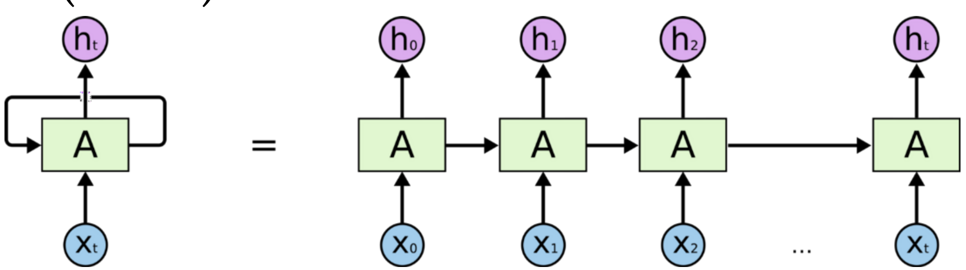
  [action searching]

Machine-learned features

- Learning features automatically
- Requires a large amount of (labelled) training data
- Successful approaches based on CNN, RNN, LSTM
  - Convolutional neural networks (CNN)



  - Recurrent neural networks (RNN)



Existing ML approaches

- 16–256D float vectors compared by the Euclidean distance

  [Coskun et al.: Human Motion Analysis with Deep Metric Learning. ECCV, 2018]
  [action recognition] [gait recognition]

- ?D float vectors compared by the Euclidean distance

  [Aristidou et al.: Deep Motifs and Motion Signatures, ACM Trans. Graph., 2018]
  [action recognition] [action searching]

- 4,096D float vectors compared by the Euclidean distance

  [Sedmidubsky et al.: Effective and efficient similarity searching in motion capture data. MTAP, 2018]
  [action recognition] [action searching]

- 160D bit vectors compared by the Hamming distance

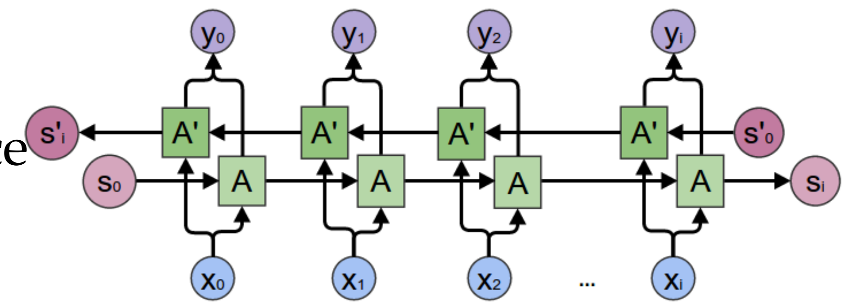
  [Wang et al.: Deep signatures for indexing and retrieval in large motion databases. Motion in Games, 2015]
  [action searching]

Advantages/disadvantages of features

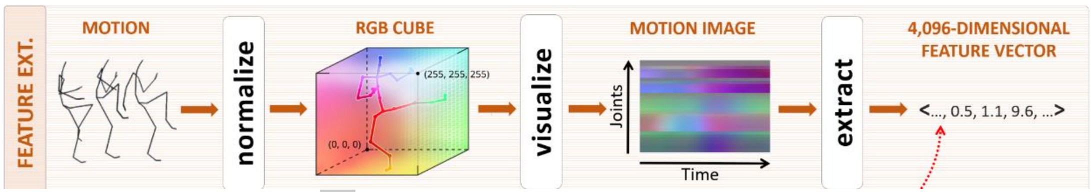
	HAND-CRAFTED	MACHINE-LEARNED
Accuracy (descriptive power)		
Interpretability of dimensions		
Prerequisites	Very good scenario knowledge	Many example categorized motions
Application	More-easily describable scenarios	Most scenarios with some categorization

Machine-learning approaches in detail

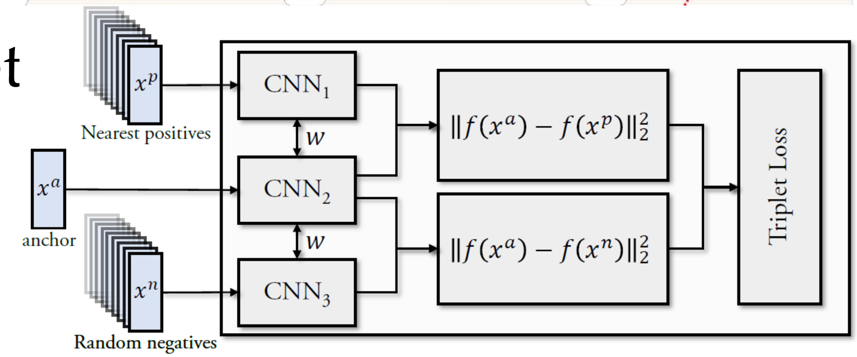
- LSTM-based Similarity Concept
  - 1,024D feature + Manhattan distance



- Motion-Image Similarity Concept
  - 4,096D feature + Euclidean distance



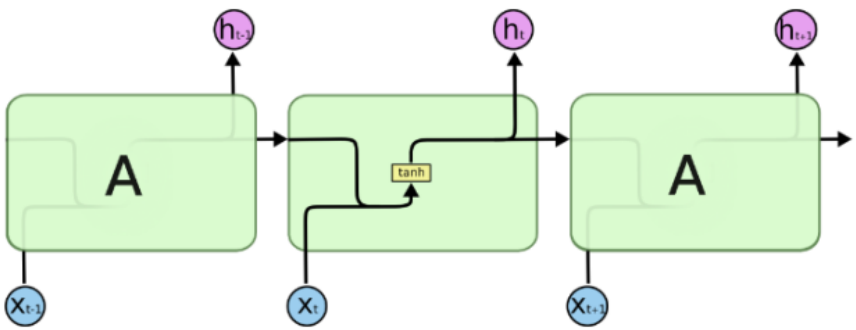
- Triple-Loss Similarity Concept
  - ?D feature + Euclidean distance

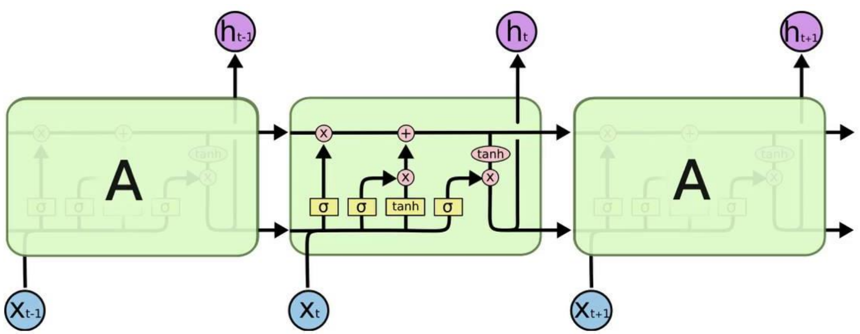


4.3 LSTM-based Similarity Concept

Recurrent Neural Networks (RNN)

- Ideal to model sequences of poses
- Output contents are influenced by the history of inputs

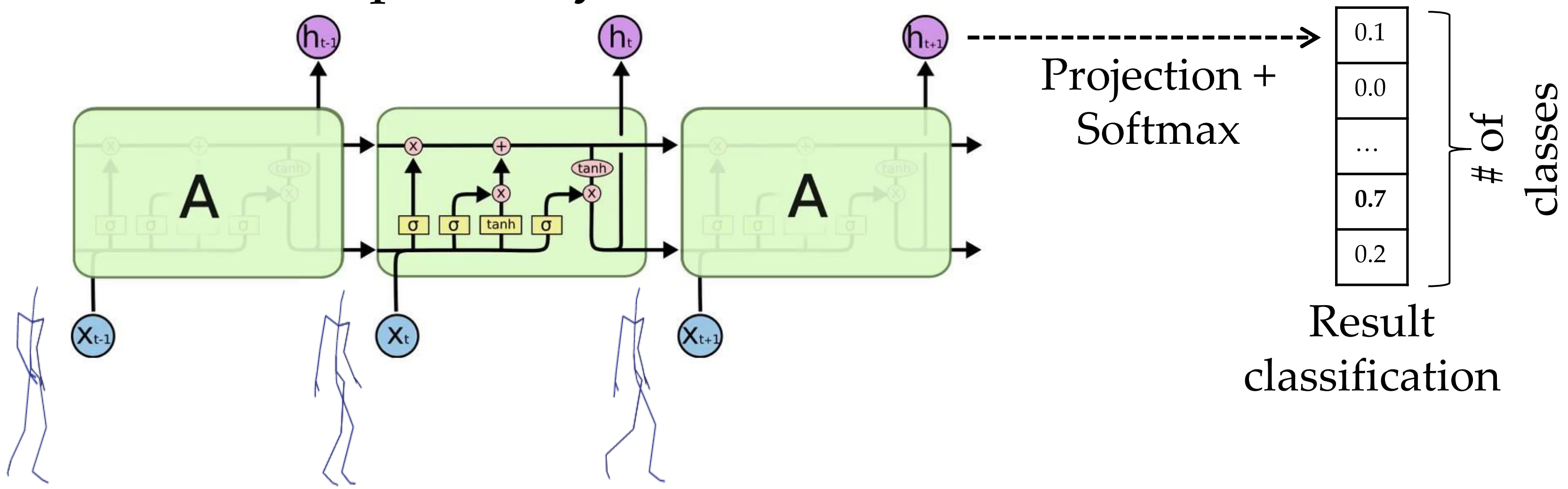




- **Long-Short Term Memory (LSTM)** network:
  - A special kind of RNN, capable of learning long-term dependencies
  - It learns when data should be remembered and when they should be thrown away

LSTM-based similarity measure (LSTM features)

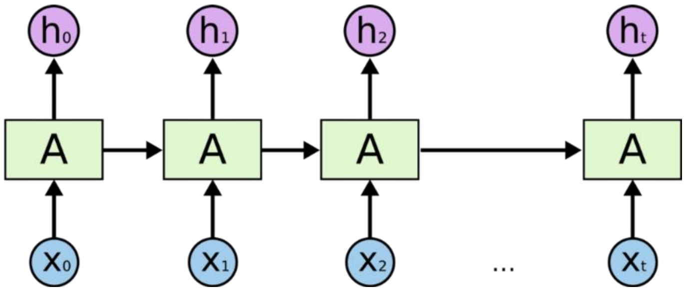
- Number of states/cells corresponds to the number of poses
- The last state h_{t+1} can be used as a feature
- Size of each state h_i is a user-defined parameter
  - Suitable state size of 512 / 1,024 / 2,048 dimensions
- Features compared by the Manhattan/Euclidean distance



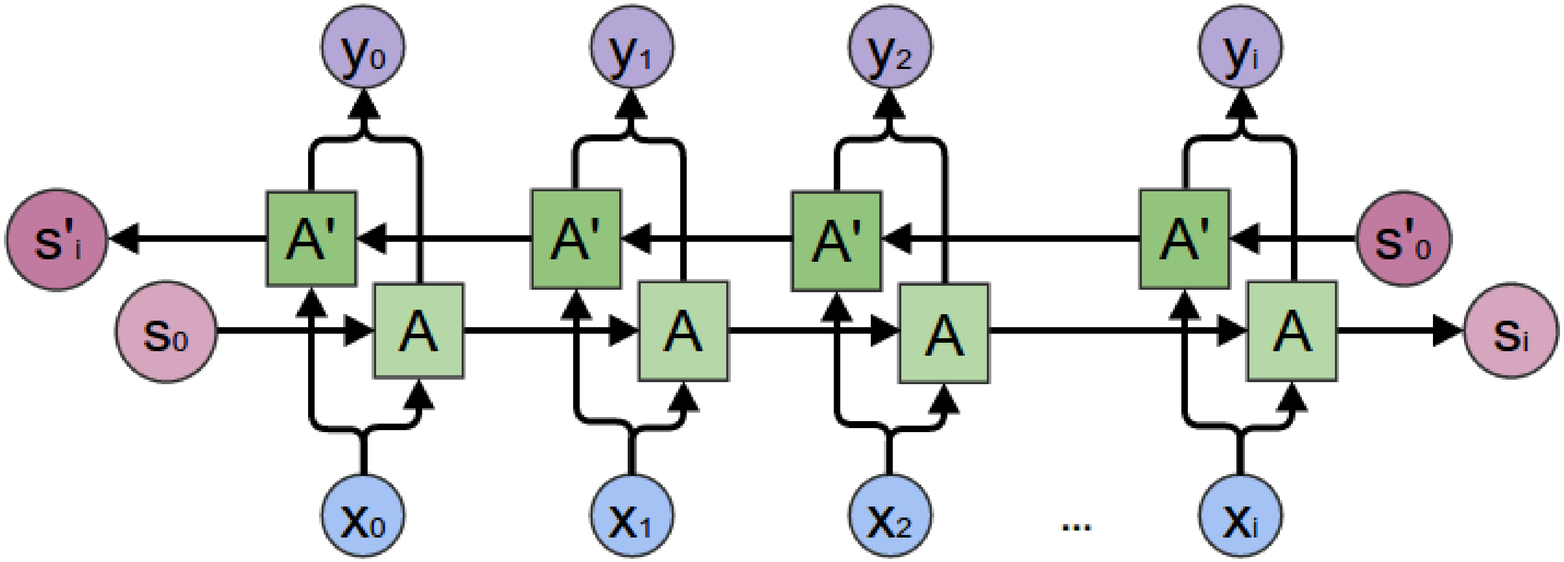
ACM ICMR 2019 Ottawa

LSTM-based similarity measure (LSTM features)

- Standard LSTM architecture:



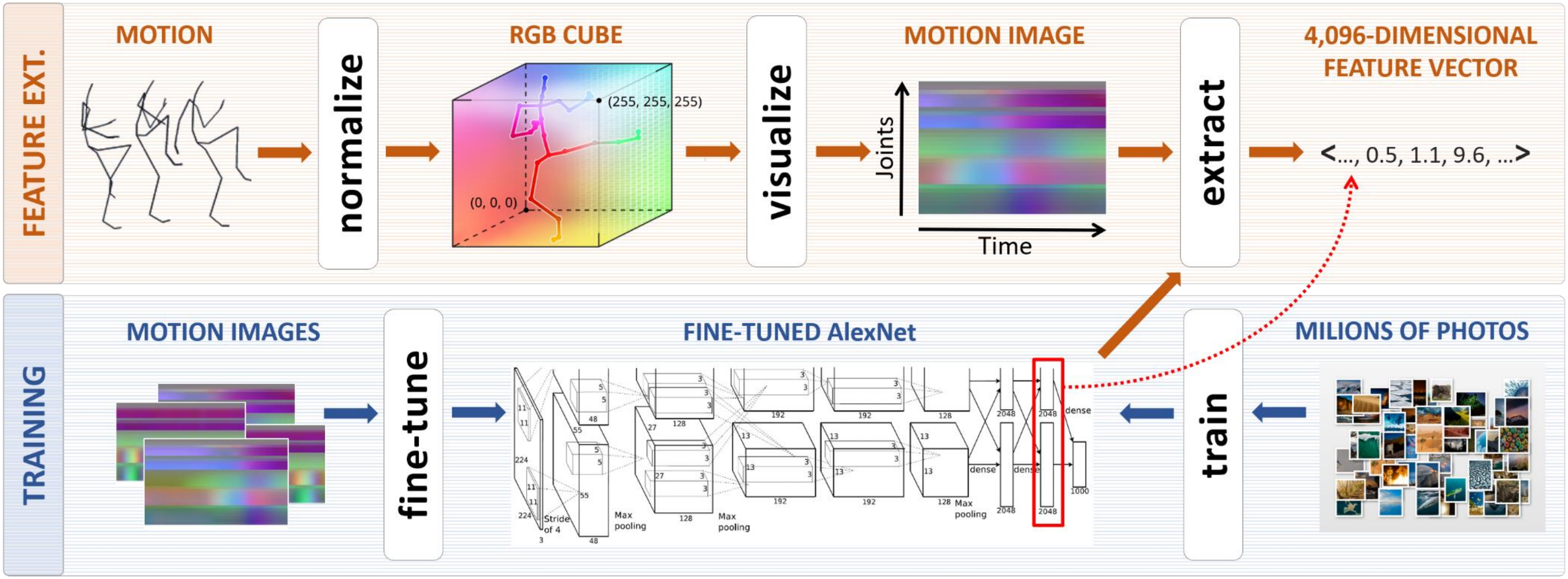
- Bidirectional LSTM extension:



Motion-image similarity measures (CNN features)

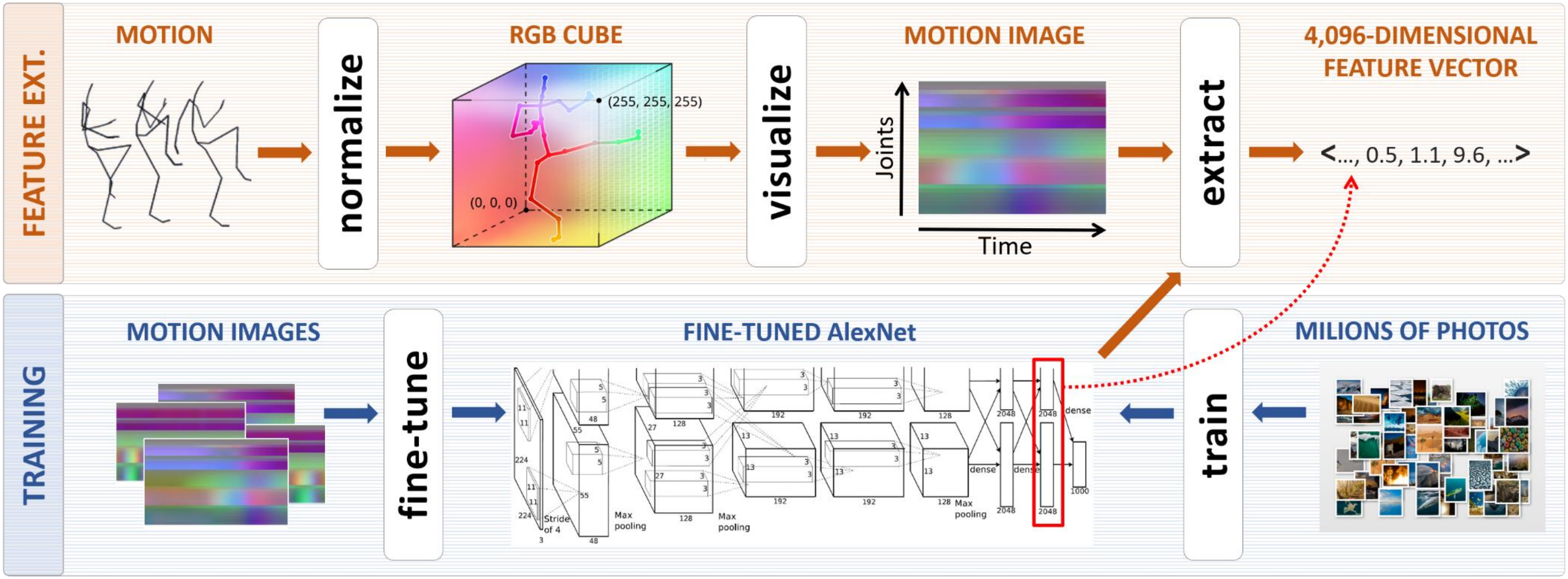
[Sedmidubsky et al.: Effective and efficient similarity searching in motion capture data. Multimedia Tools and Appl., 2018]

- Deep 4,096D features compared by the Euclidean distance
- Suitable for motions in order of seconds (~ actions)



Feature extraction steps

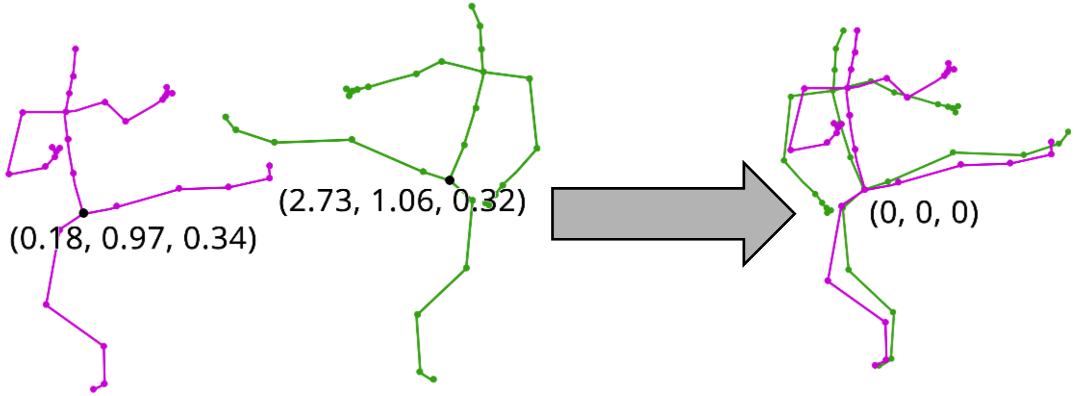
1) Normalizing motion data (optional context-dependent step)
2) Transforming normalized data into a 2D motion image
3) Extracting a 4,096D feature from the image using a DCNN



4.4 Feature Extraction – Normalization

Feature extraction steps

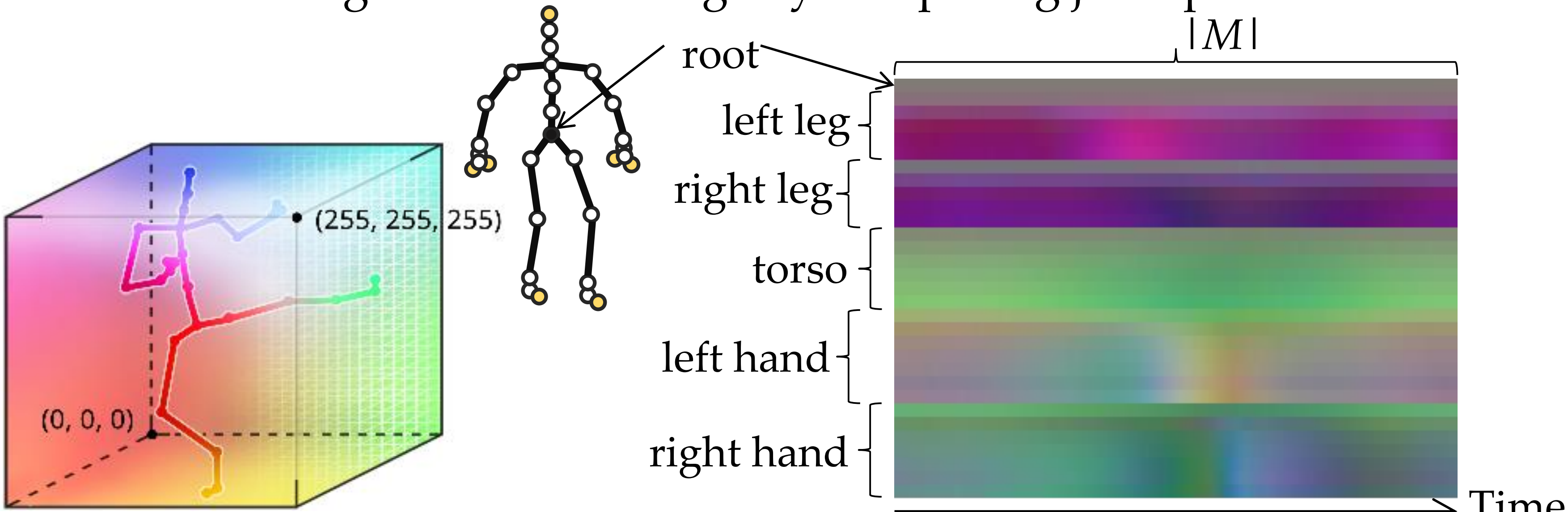
1) Normalizing motion data
   - Optional step – its utilization depends on a target application
   - Normalizing each pose independently vs. conditionally
   - E.g., position, orientation, and skeleton-size normalization in each pose independently is suitable for classifying daily activities



4.4 Feature Extraction – Visualization

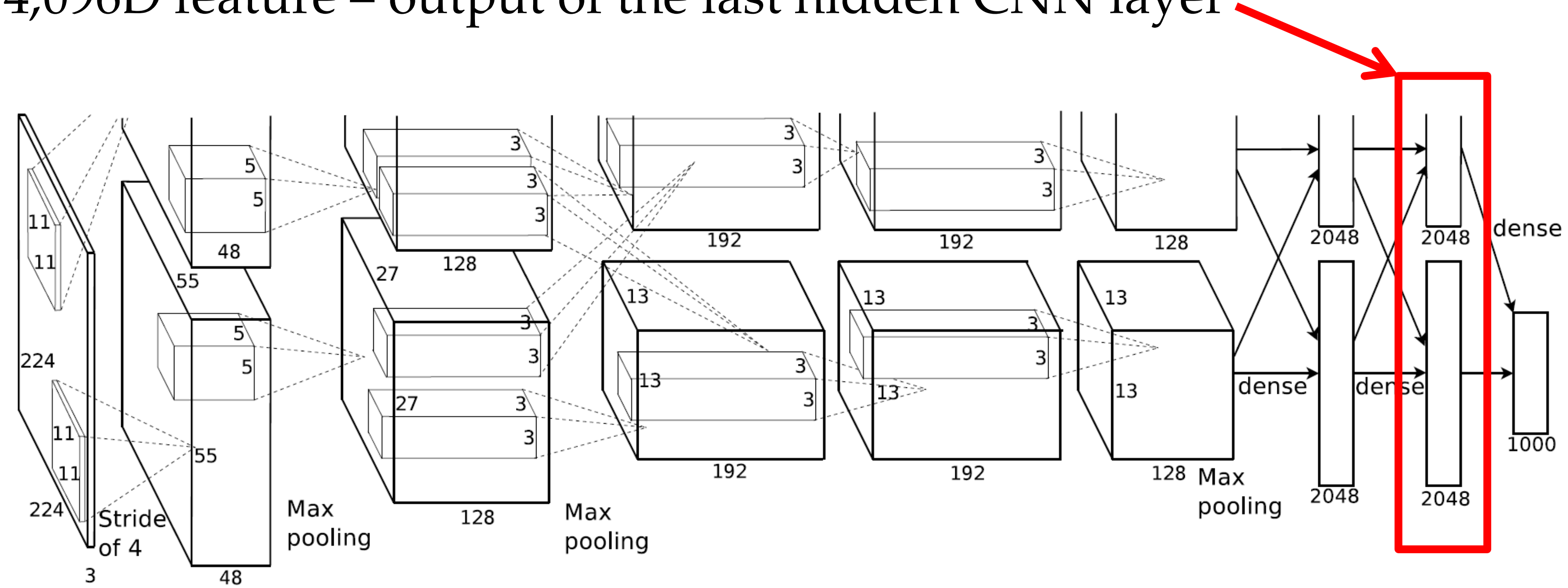
Feature extraction steps

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



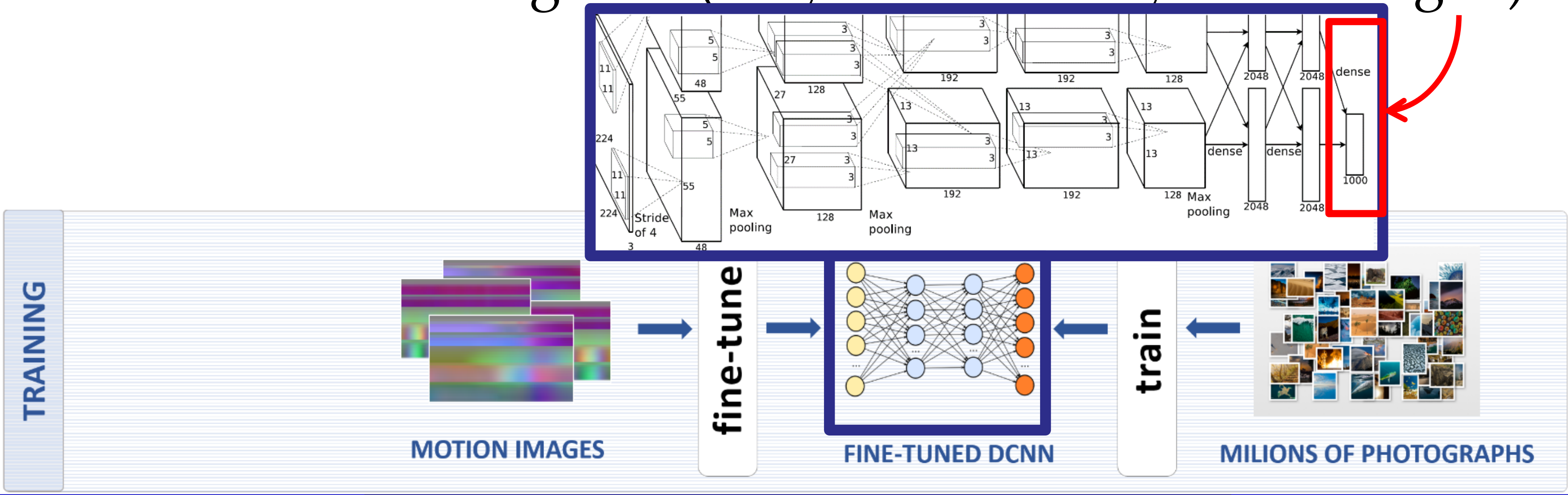
Feature extraction steps

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



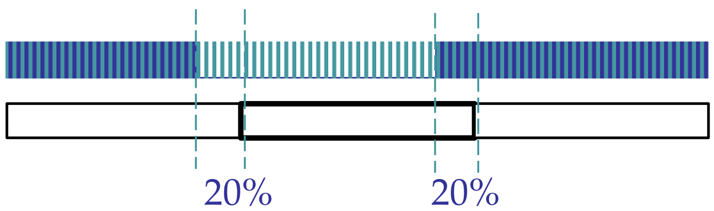
Fine-tuning the CNN ~ transferred learning

- 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)

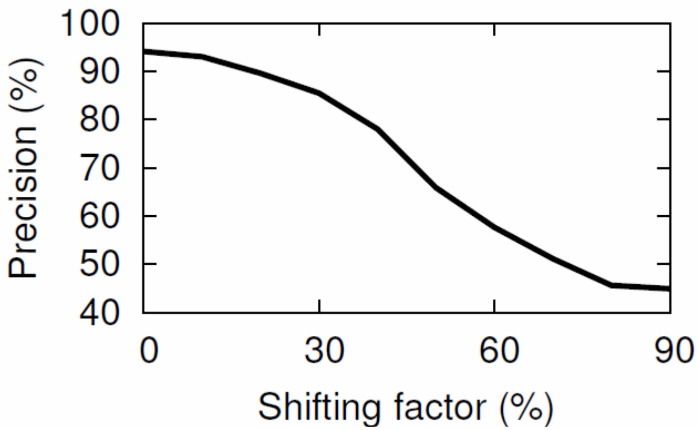


Elasticity property

- 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 CNN
  - 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
  - Even motions of categories that have not been available during the training phase are well clustered

Triple loss similarity concept

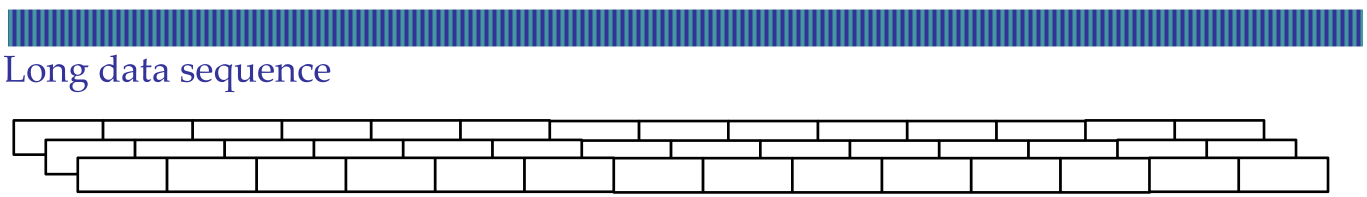
[Aristidou et al.: Deep Motifs and Motion Signatures, ACM Transactions on Graphics, 2018]

- Learning features in an unsupervised way, i.e., without labelled training data
- Requirements:
  - A granular similarity concept to determine very similar and very dissimilar motions
  - A large amount of "triples" of training data
    - Triplet = anchor + its similar (positive) motion + its dissimilar (negative) motion

4.5 Triple Loss Similarity Concept

Triple loss similarity concept

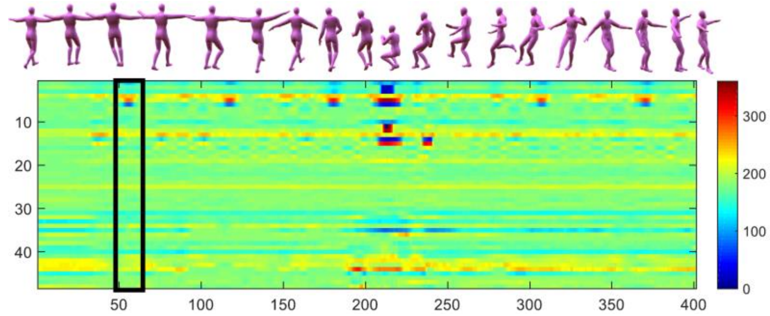
- Triplets generated from long motion sequences by a synthetic segmentation technique
  - Segments of about 0.7s with a shift of about 0.15s



- Triplets:
  - Anchor – randomly selected segment
  - Positive examples – segments temporarily close to the anchor with no overlap or matched using DTW on joint-angle rotations
  - Negative examples – segments temporally far away from the anchor or taken from another motion sequence

Triple loss similarity concept

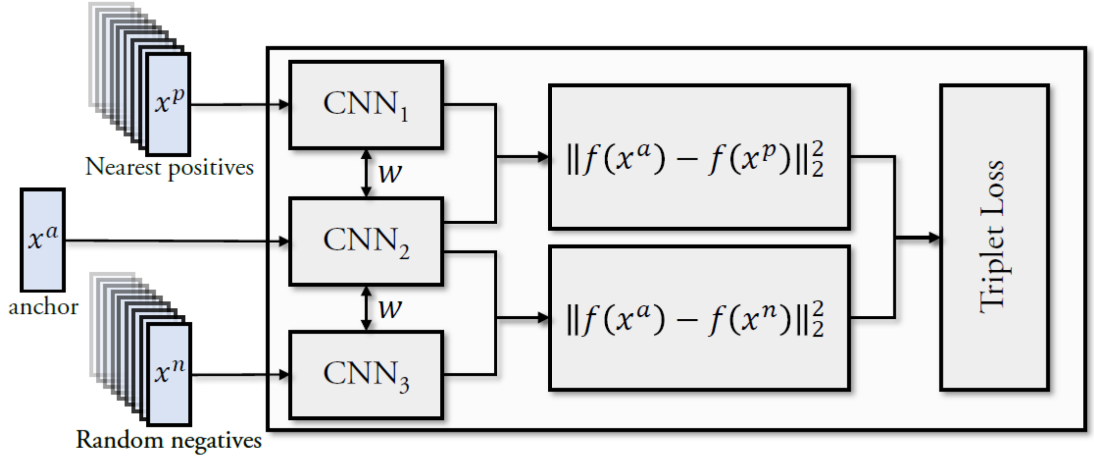
- Three CNNs sharing the same parameters
- Inputs in form of motion images



- Loss function:

$$L(x^a, x^p, x^n) = [\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + \alpha]_+$$

- Architecture:

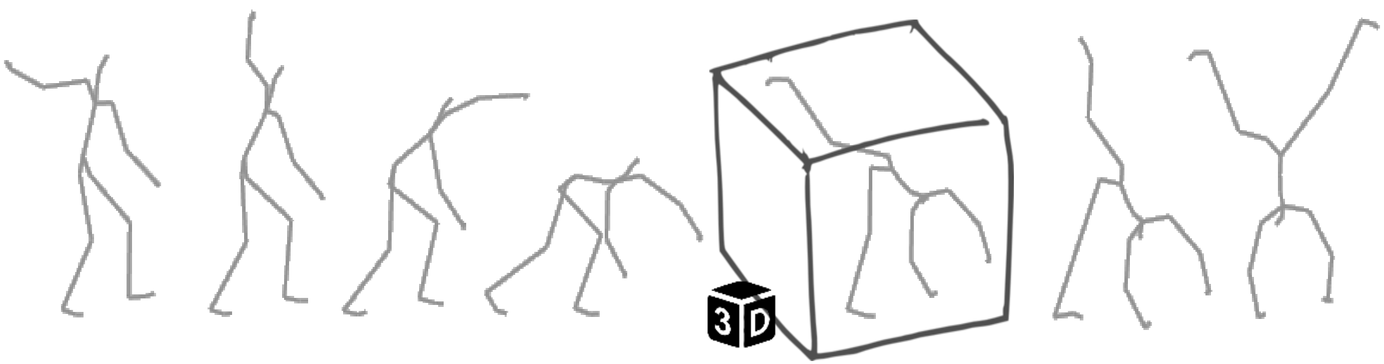


Advantages/disadvantages of the similarity concepts

	LSTM-based	CNN-based	Triple-loss-based
Accuracy (descriptive power of features)	🙂	🙂	🙂
Volume of training data	🙂	🙂	🙁
Input data preprocessing	🙂	😐	🙁
Length of motions	🙂	😐	🙂
Feature-size flexibility	🙂	🙁	🙂
Labelled data needed	🙁	🙁	🙂

5 Metric Searching as a Data-Access Paradigm

5.1 Similarity Queries & Partitioning Principles
5.2 Indexing Structures (M-Tree, D-Index, Sketches)
5.3 Summary & Live Demo



5.1 Metric Space – Search Problem

ACM ICMR 2019 Ottawa

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
- Implementation problems:
  - How to partition the data to reduce search space
  - How to ask questions – definition of queries
  - How to execute queries – to achieve performance
- The challenge: in metric spaces, no total ordering exists!

5.1 Metric Space – Partitioning Principles

ACM ICMR 2019 Ottawa

Basic partitioning principles

- Given a set $X \subseteq \mathcal{D}$ in metric space $\mathcal{M} = (\mathcal{D}, d)$, basic partitioning principles have been defined:
  - Ball partitioning
  - Generalized hyper-plane partitioning

- Note:
  - Some special cases, such as Euclidian or Supermetric spaces are more tractable

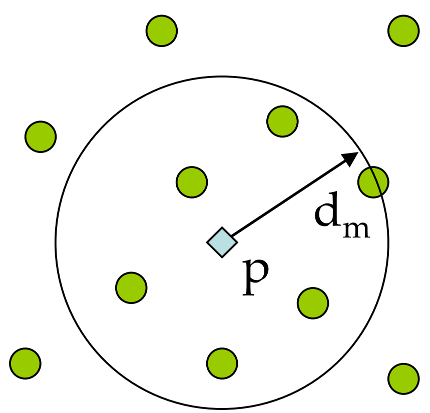
5.1 Metric Space – Partitioning Principles

Basic partitioning principles

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

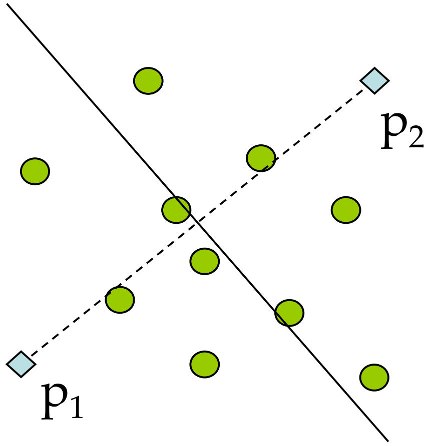
Ball partitioning

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

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

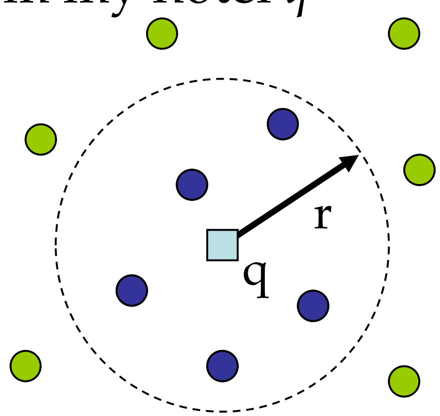


5.1 Metric Space – Similarity Queries

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

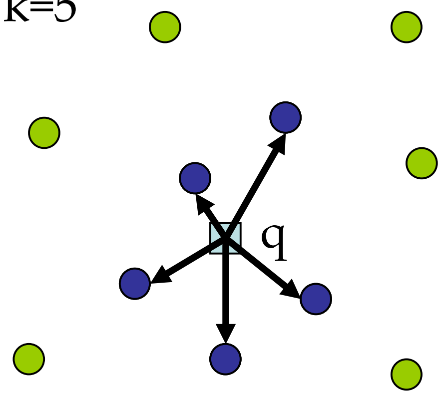
$NN(q) = \{x \in X \mid \forall y \in X, d(q,x) \leq d(q,y)\}$

k-nearest neighbor query

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

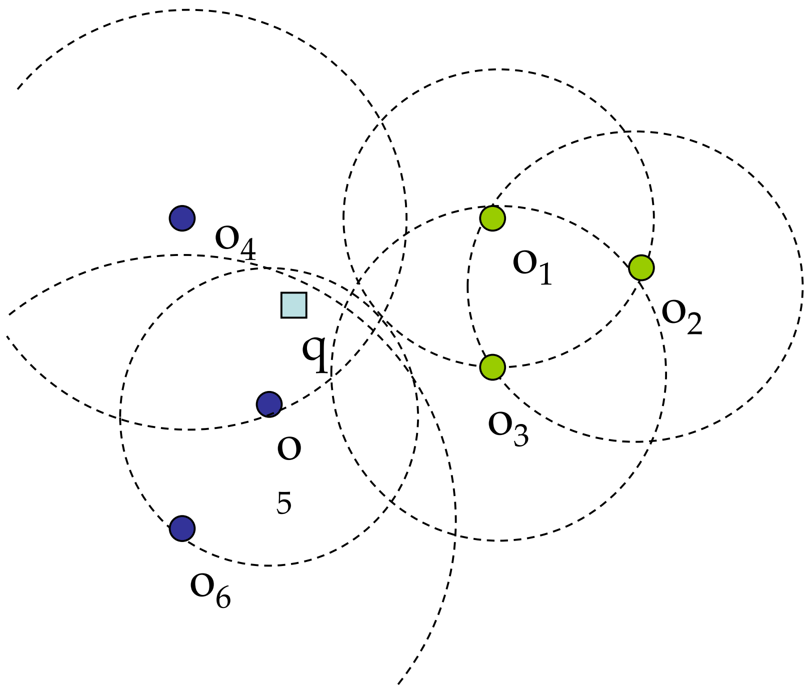
"five closest museums to my hotel *q*"

k=5



Reverse nearest neighbor query

$$k\text{-}RNN(q, k) = \{R \subseteq X, \\ \forall x \in R: q \in k\text{-}NN(x, k) \wedge \\ \forall x \in X - R: q \notin k\text{-}NN(x, k)$$



"all hotels with a specific museum as a nearest cultural heritage cite"

Example of 2-*RNN*: objects o_4, o_5, and o_6 have q between their two nearest neighbors

Similarity joins

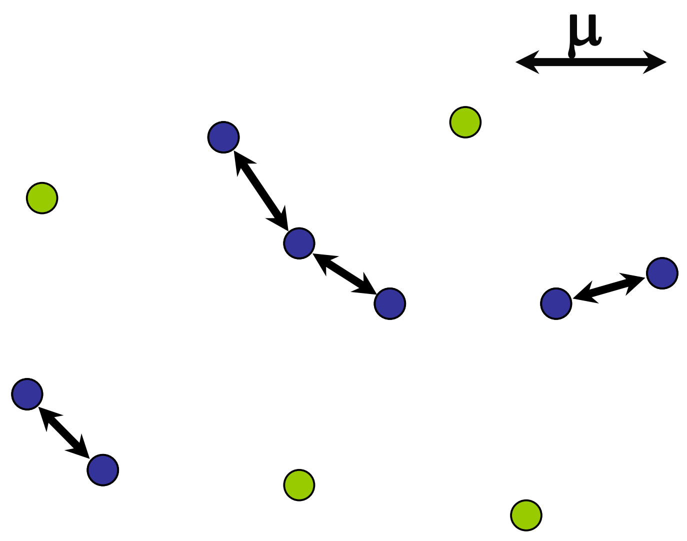
- Similarity join of two data sets

$$X \subseteq \mathcal{D}, Y \subseteq \mathcal{D}, \mu \geq 0$$

$$J(X, Y, \mu) = \{(x, y) \in X \times Y : d(x, y) \leq \mu\}$$

- Similarity self join $\Leftrightarrow$ $X = Y$

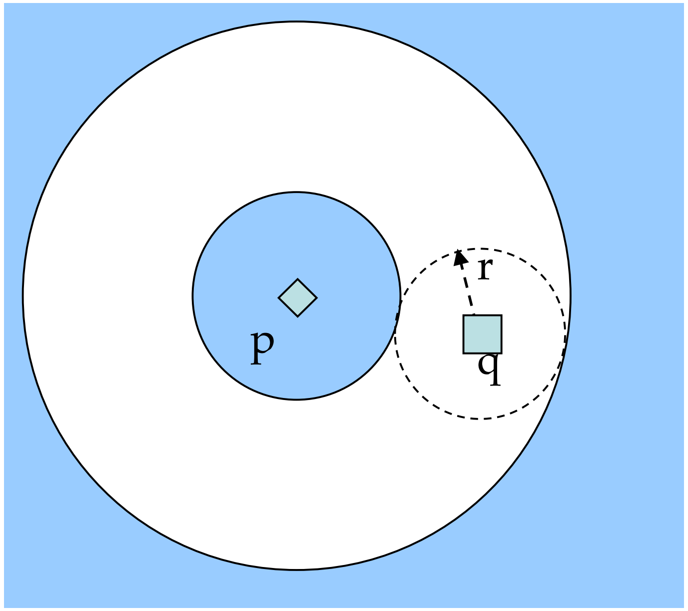
“pairs of hotels and museums which are five minutes walk apart”



Pivot filtering

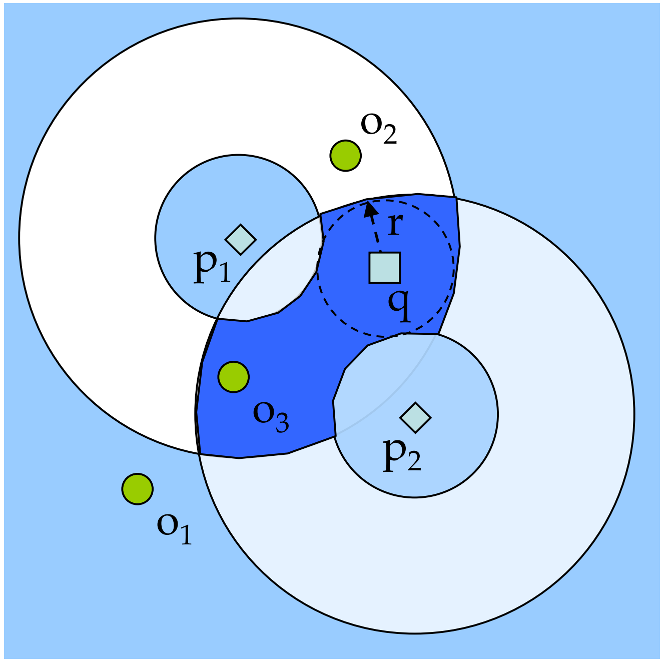
- Idea: Given $R(q, r)$, use triangle inequality for pruning
- All distances between objects and a pivot p are known
- Prune object $o \in X$ if any holds:

$$d(p,o) < d(p,q) - r$$
$$d(p,o) > d(p,q) + r$$



Pivot filtering

- Filtering with two pivots:
  - Only objects in the dark blue region have to be checked
  - Effectiveness is improved using more pivots



5.1 Metric Space – Pivot Filtering

ACM ICMR 2019 Ottawa

Pivot filtering summary

- Given a metric space $\mathcal{M} = (\mathcal{D}, d)$ and a set of pivots $P = \{p_1, p_2, p_3, \ldots, p_n\}$, define a mapping function Ψ: $\Psi: (\mathcal{D}, d) \rightarrow (R^n, L_\infty)$ as:

$$\Psi(o) = (d(o, p_1), \ldots, d(o, p_n))$$

- Then, we can bound the distance $d(q, o)$ from:

$$L_\infty(\Psi(o), \Psi(q)) \leq d(q, o)$$

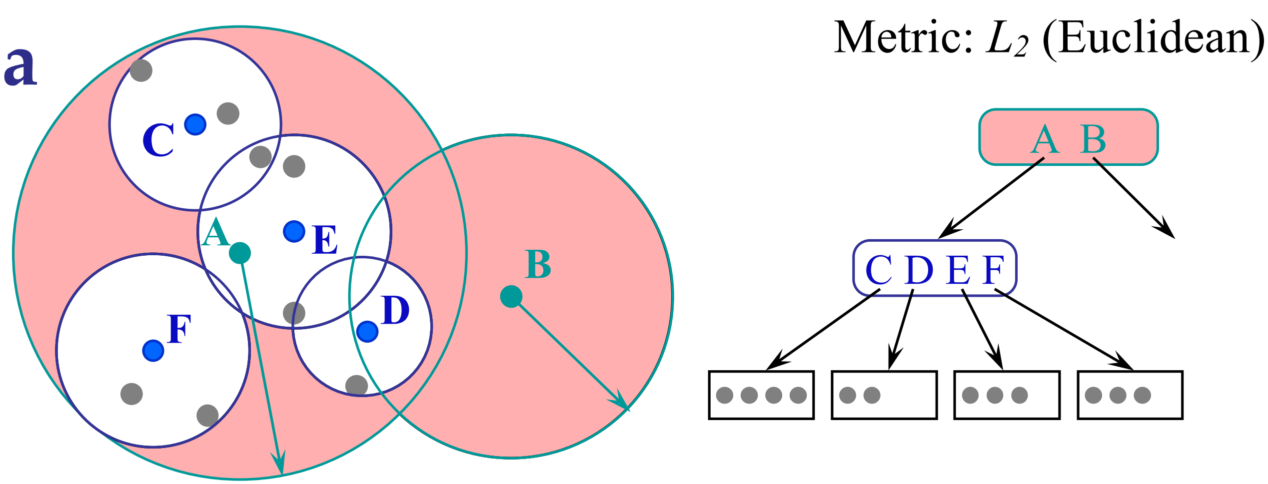
M-tree

[Ciaccia P., Patella M., and Zezula P.: M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. VLDB, 1997]

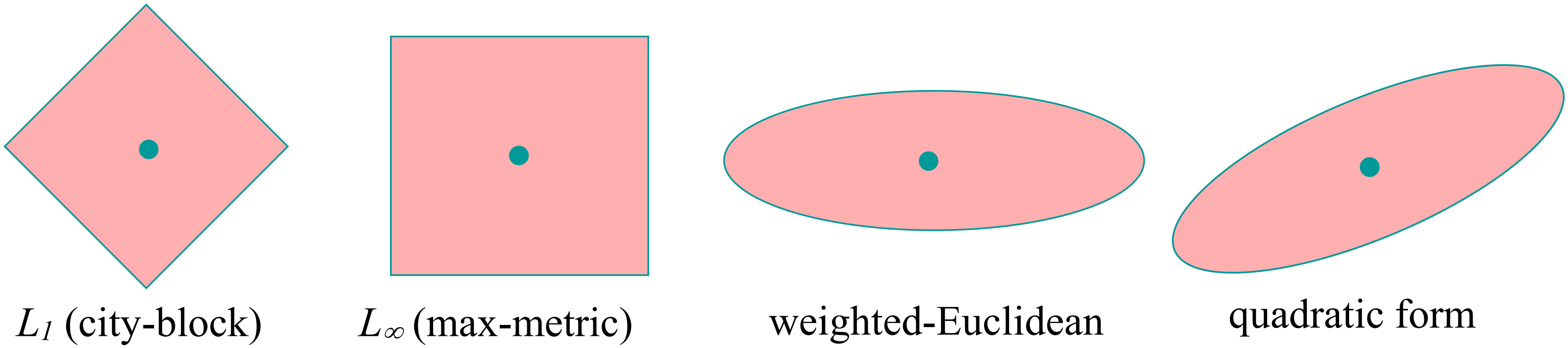
1) Paged organization
2) Dynamic
3) Suitable for arbitrary metric spaces
4) I/O and CPU optimization – computing d can be time-consuming

ACM ICMR 2019 Ottawa

The M-tree idea

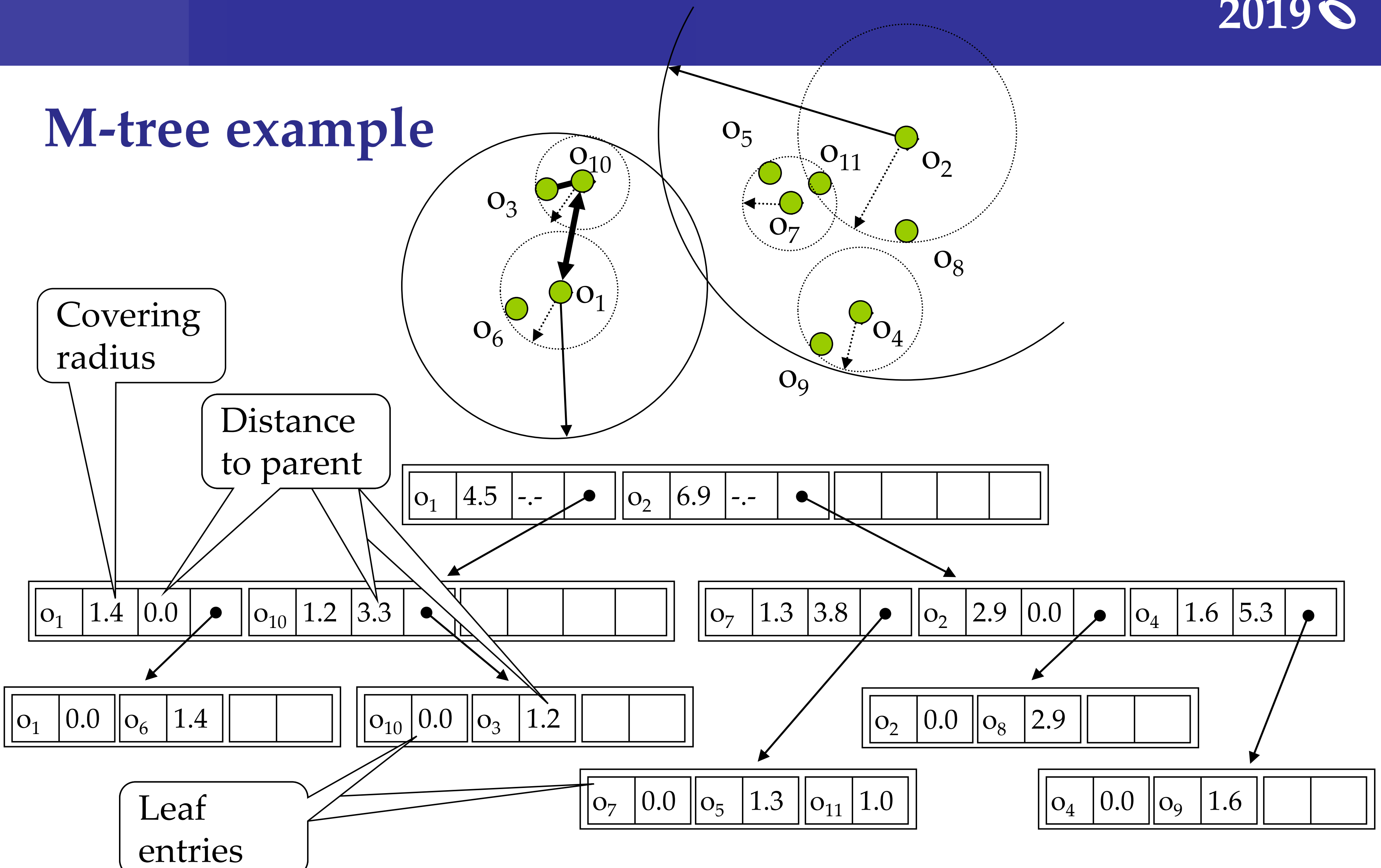


- Depending on the metric, the "shape" of index regions changes



ACM ICMR 2019 Ottawa

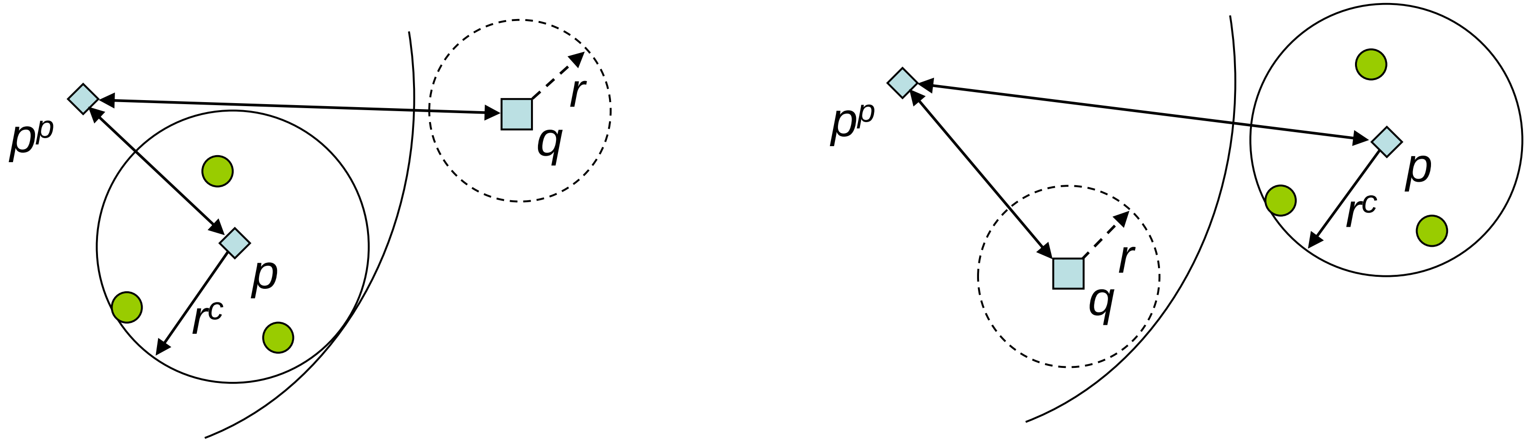
M-tree example



ACM ICMR 2019 Ottawa

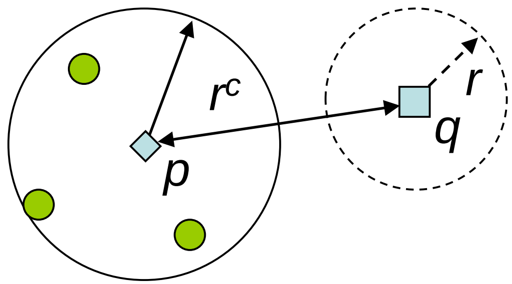
M-Tree – range search

- Given $R(q, r)$:
  - Traverse the tree in a depth-first manner
  - In an internal node, for each entry $\langle p, r^c, d(p, p^p), ptr \rangle$:
    - Prune the subtree if $|d(q, p^p) - d(p, p^p)| - r^c > r$
    - Application of the pivot-pivot constraint



M-Tree – range search

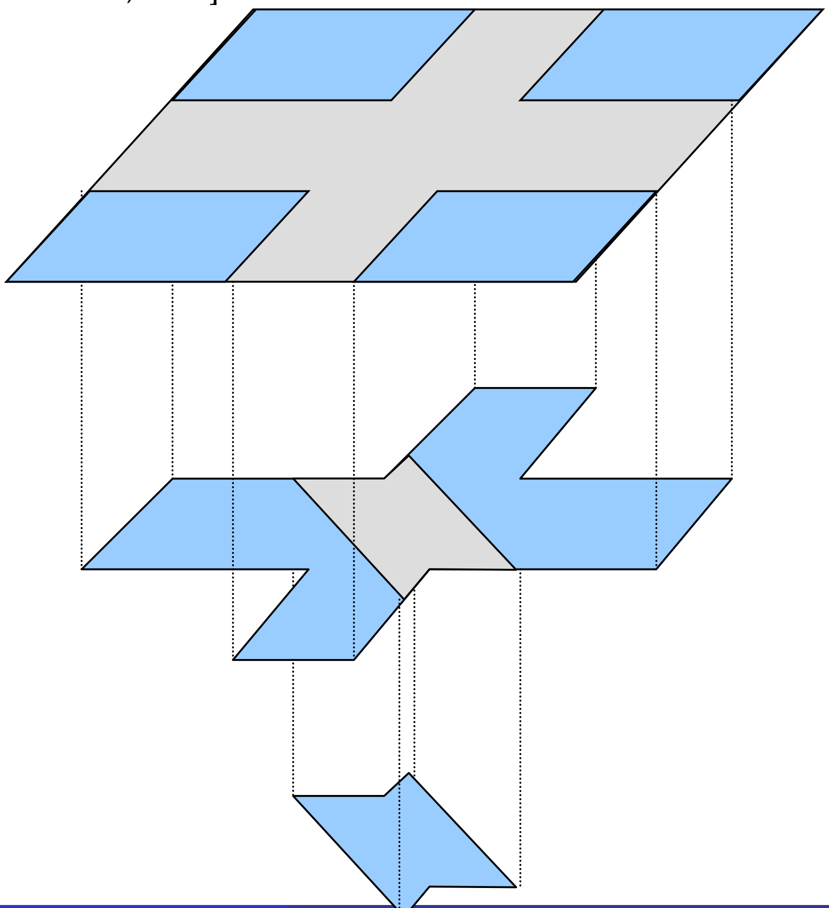
- If not discarded, compute $d(q, p)$ and
  - Prune the subtree if $d(q, p) - r^c > r$
  - Application of the range-pivot constraint



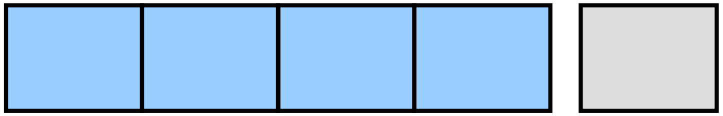
- All non-pruned entries are searched recursively

D-Index

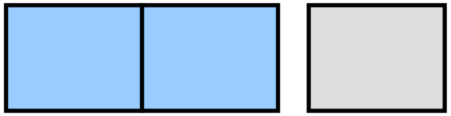
[Dohnal V., Gennaro C., Pasquale S., Zezula P.: D-Index: Distance Searching Index for Metric Data Sets. Multimedia Tools and Applications, 2003]



4 separable buckets at the first level

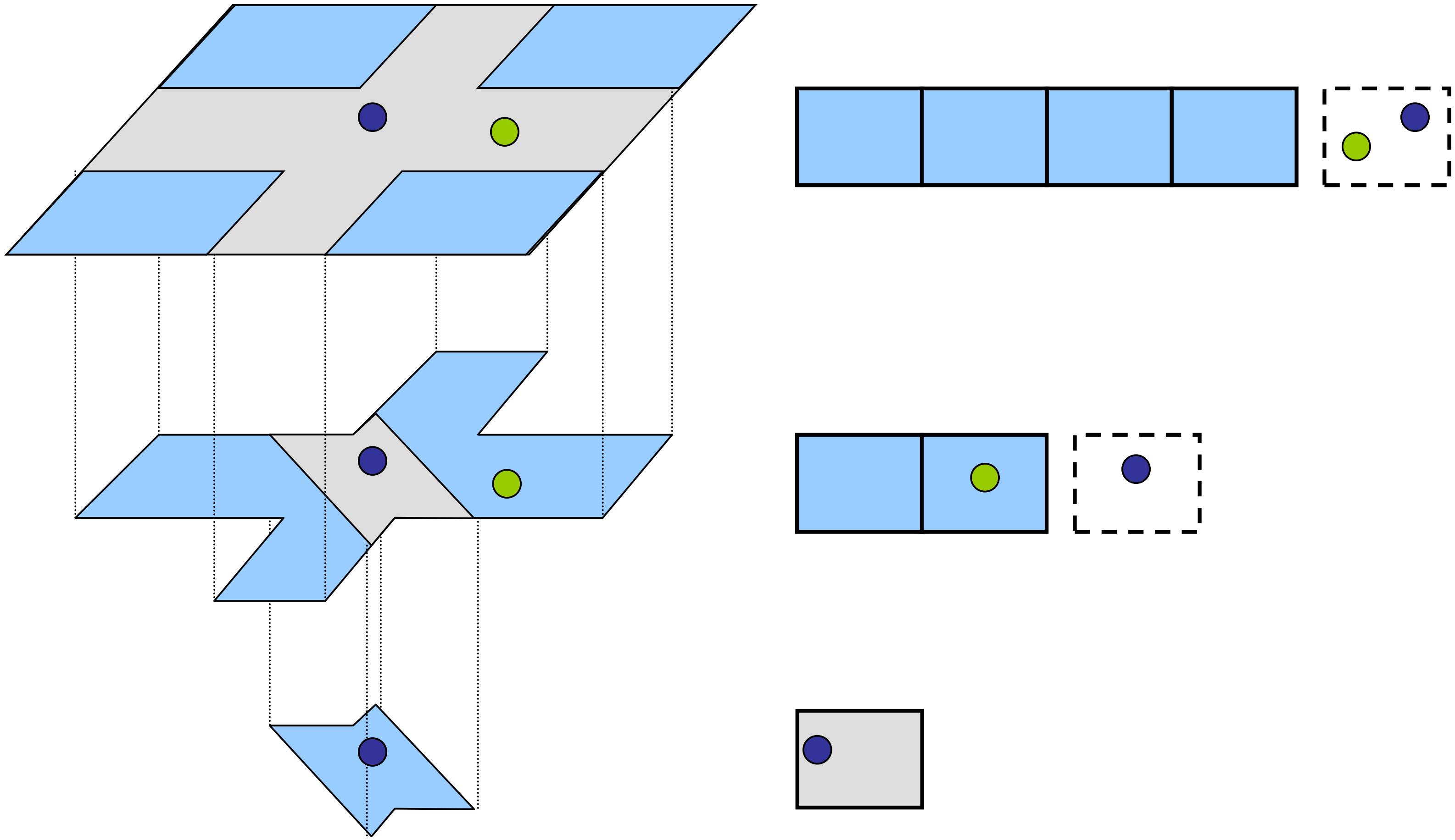


2 separable buckets at the second level



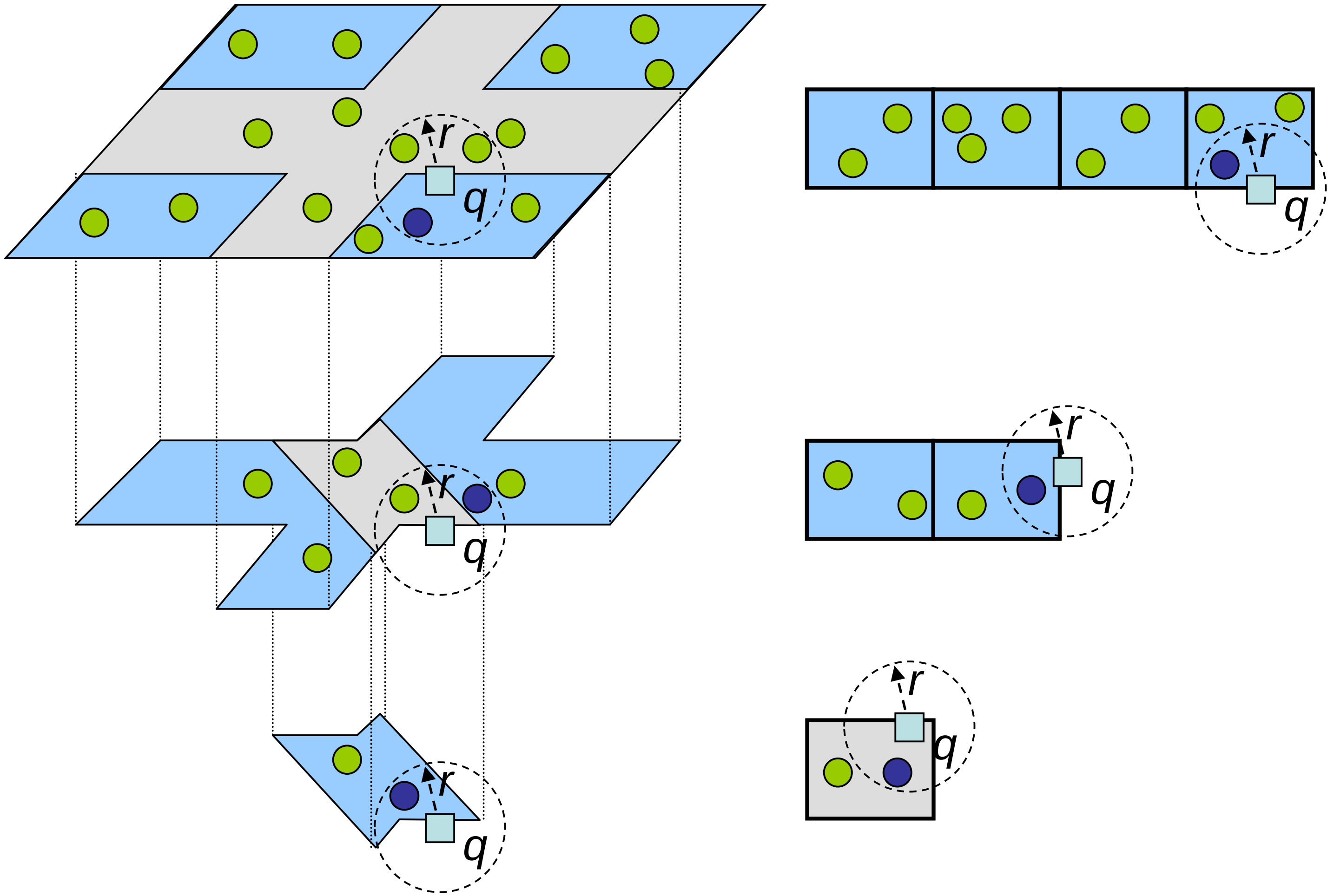
exclusion bucket of the whole structure

D-Index – insertion



ACM ICMR 2019 Ottawa

D-Index – range search



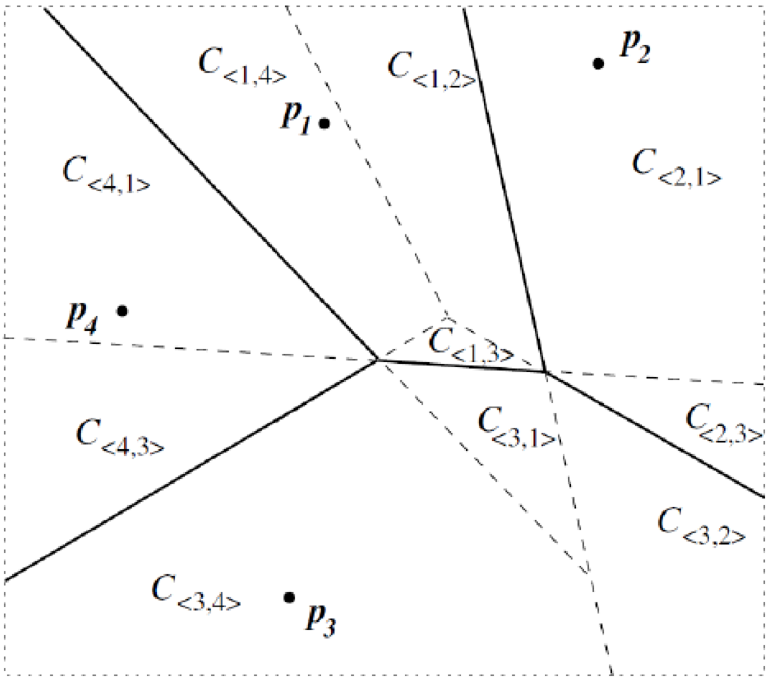
Pivot Permutation Approach

- Assume a set of n pivots $\{p_1, p_2, \ldots, p_n\}$
- Given object x in X, order the pivots according to $d(x, p_i)$
- Let $\prod_x$ be a permutation on the set of pivot **indexes** $\{1, \ldots, n\}$ such that $\prod_x(j)$ is index of the j-th closest pivot from x
  - E.g., $\prod_x(1)$ is the index of the closest pivot to x
  - $p_{\prod x(j)}$ is the j-th closest pivot from x
- $\prod_x$ is denoted as Pivot Permutation (PP) with respect to x

Pivot Permutation Approach

- Can be seen as a recursive *Voronoi* partitioning to level l
- Cell $C_{<i_1, \dots i_l>}$ contains objects x for which:

  $\prod_x(1) = i_1, \prod_x(2) = i_2, \dots, \prod_x(l)=i_l$



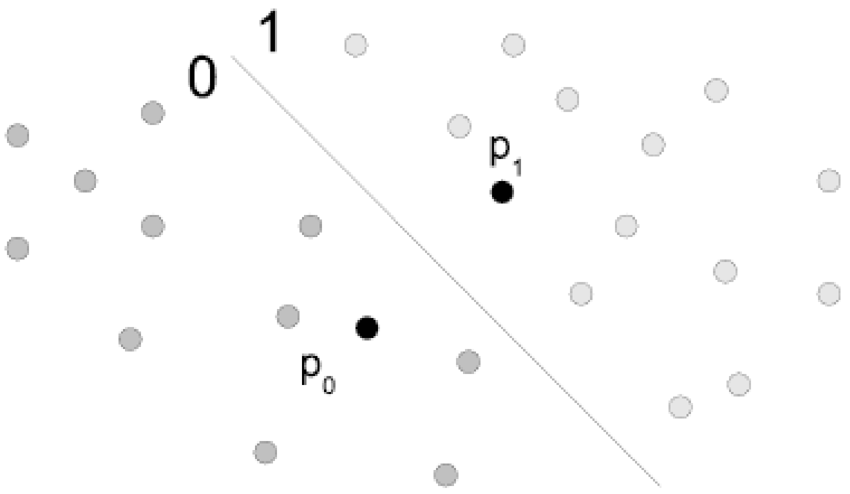
Metric sketches for fast searching and filtering

- Transformation of any metric space $(\mathcal{D}, d)$ to the Hamming space:
  - Each descriptor o is transformed into a bit-string sketch of length λ
  - Hamming distance evaluates the number of different bits – very efficient (using a hardware instruction)
- Sketches compared by the Hamming distance should approximate similarity relationships of the original metric
- Typical sketch length is 32–320 bits

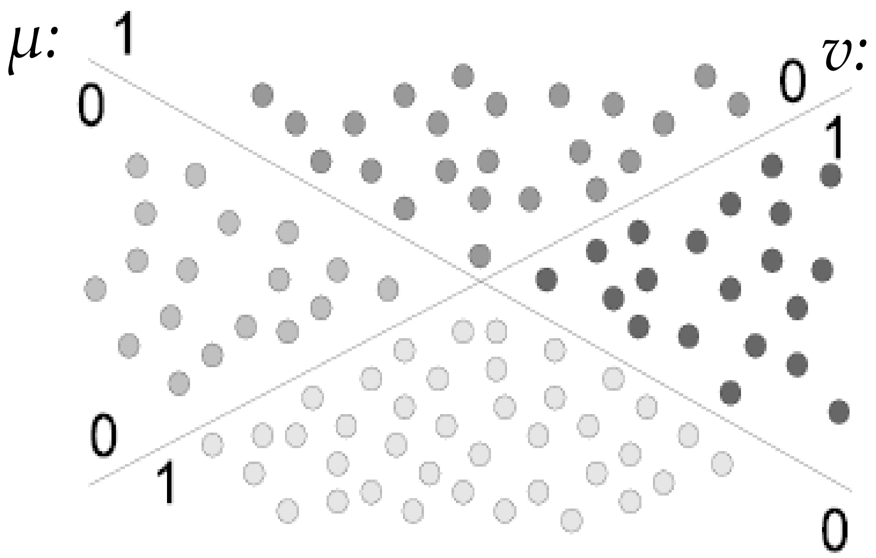
ACM ICMR 2019 Ottawa

Sketching transformation

GHP to set one bit of *sk(o)*

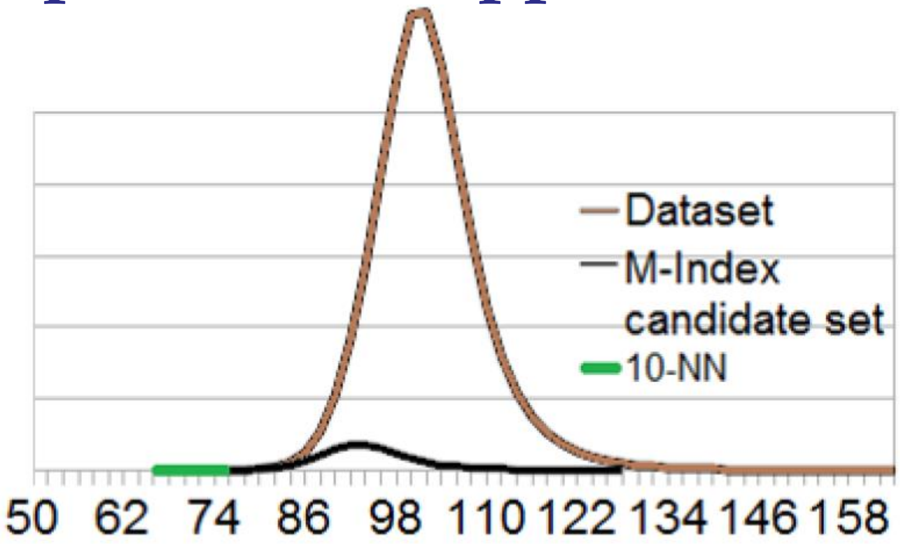


Two GHPs to set two bits μ and v of all sketches *sk(o)*



ACM ICMR 2019 Ottawa

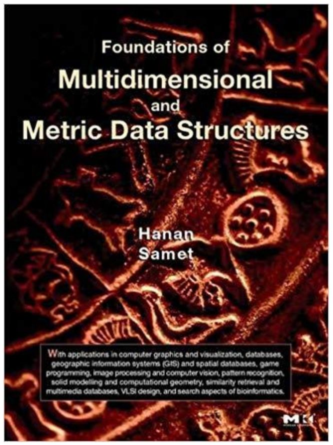
Properties and application



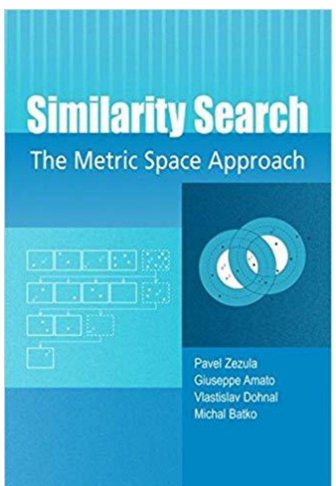
- Good sketches (choosing pivots):
  - Balanced split
  - Low correlation
- Application – even better for filtering than searching

5.3 Similarity Search Textbooks

Major textbooks on metric searching technologies



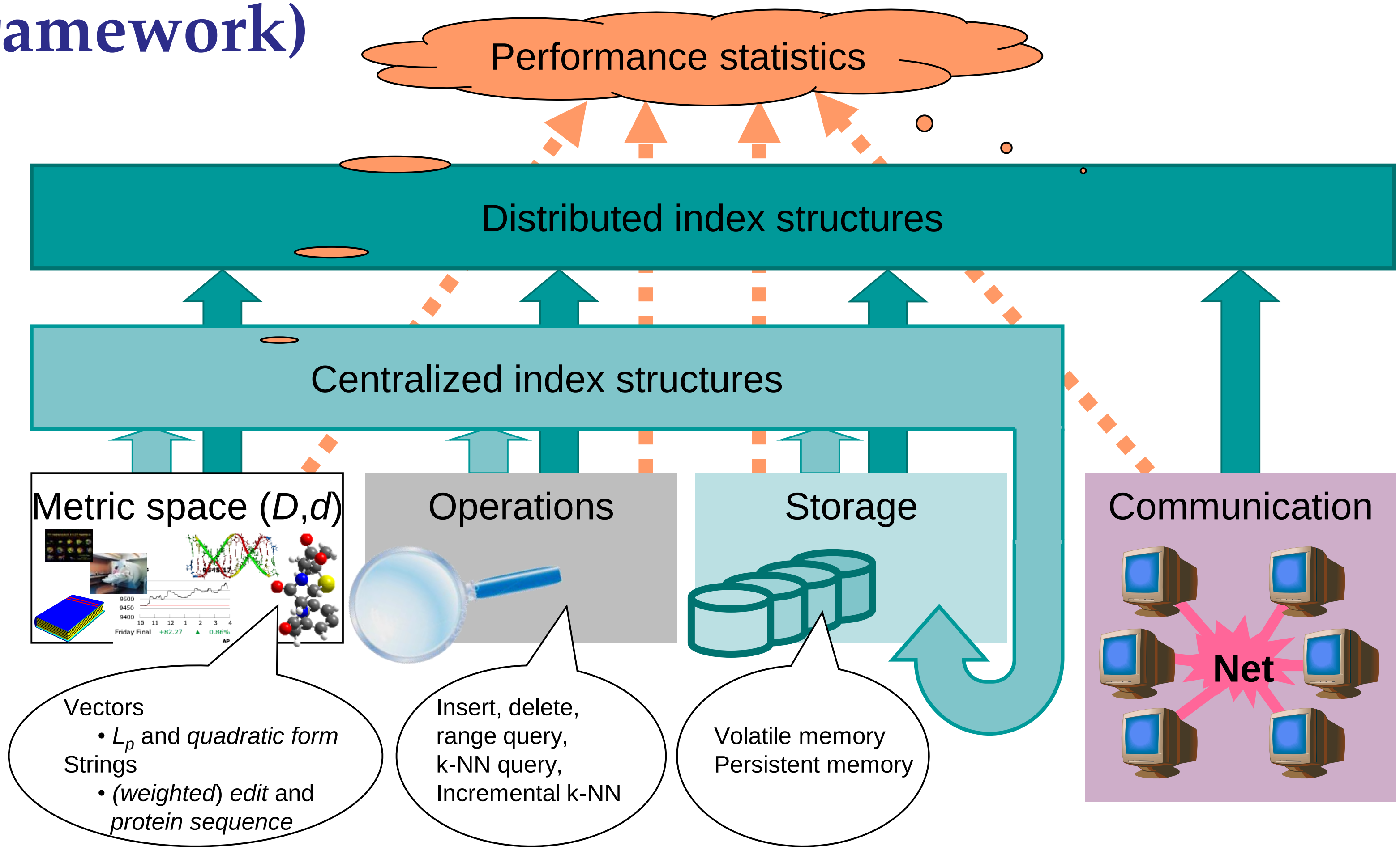
H. Samet
Foundation of Multidimensional and Metric Data Structures
Morgan Kaufmann, 1,024 pages, 2006



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

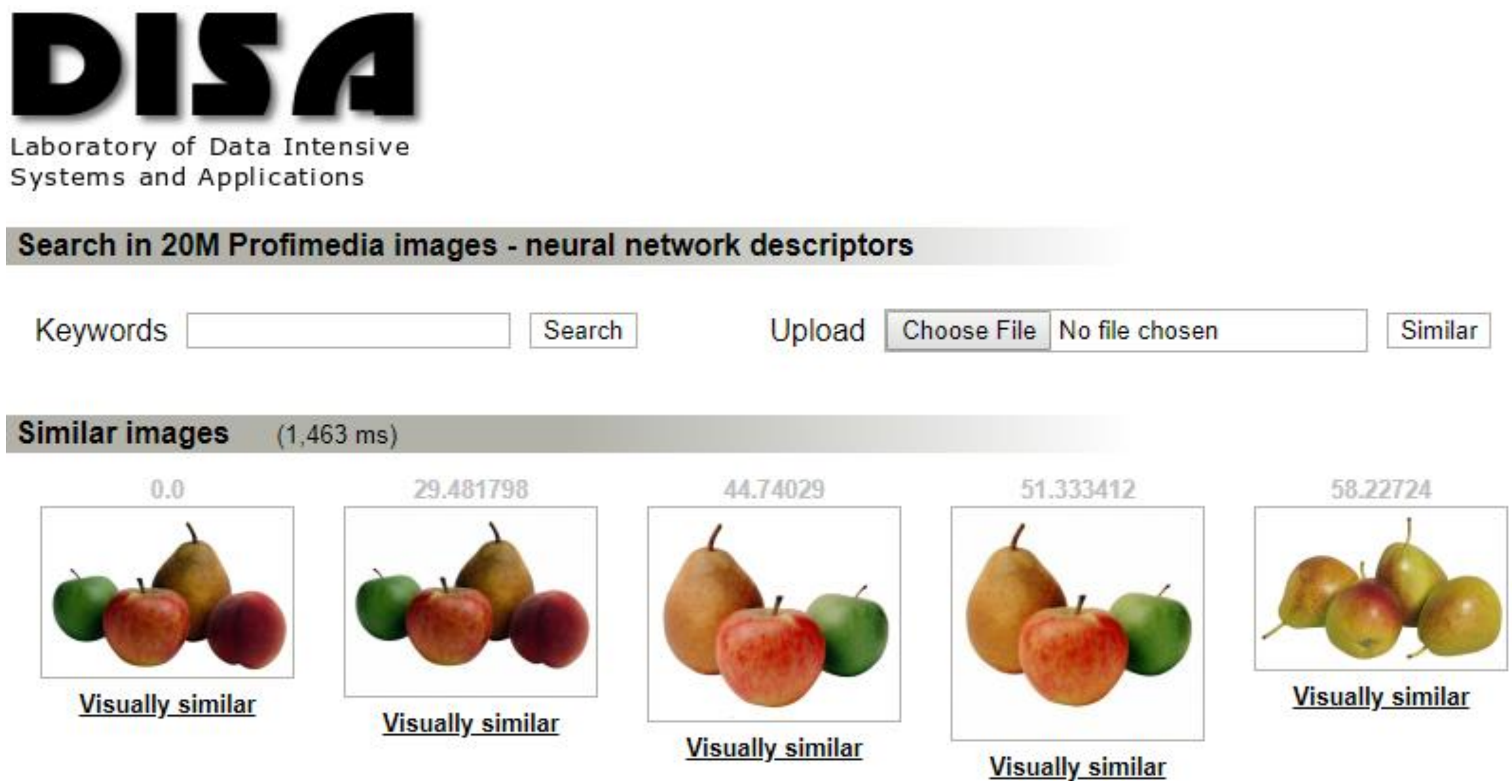
Teaching materials:
http://www.nmis.isti.cnr.it/amato/similarity-search-book/

MESSIF (Metric Similarity Search Implementation Framework)

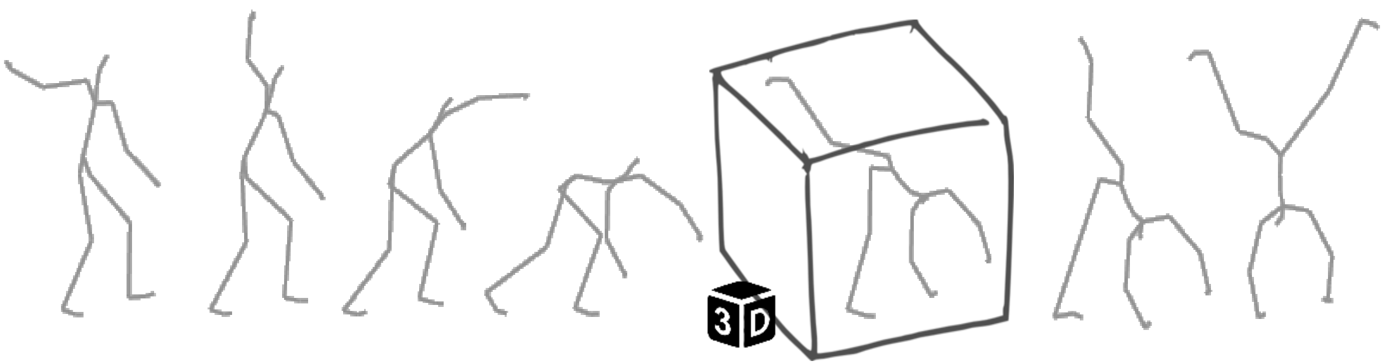


Similarity search demos – scalability

- 20M images: http://disa.fi.muni.cz/demos/profiset-decaf/



6 Action Recognition



6.1 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

6.1 Action Classification

Knowledge base

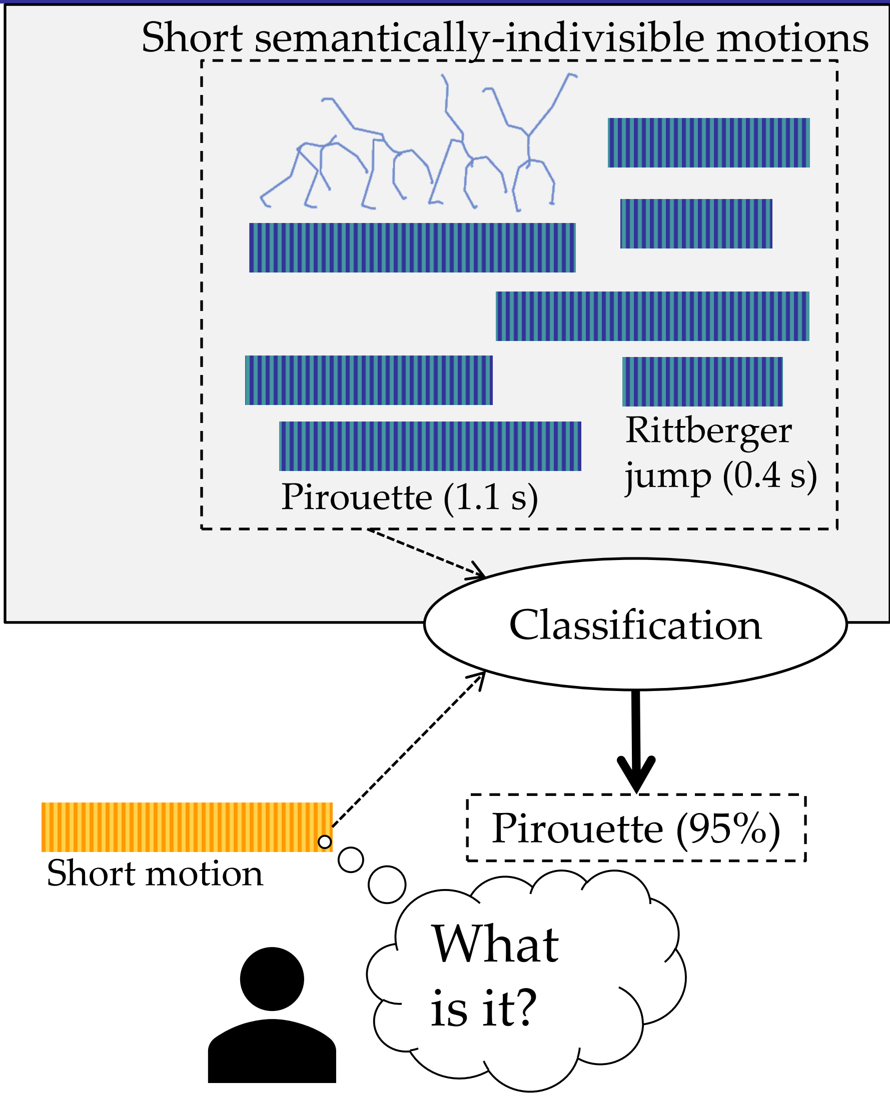
- Collection of labeled short actions ~ training data

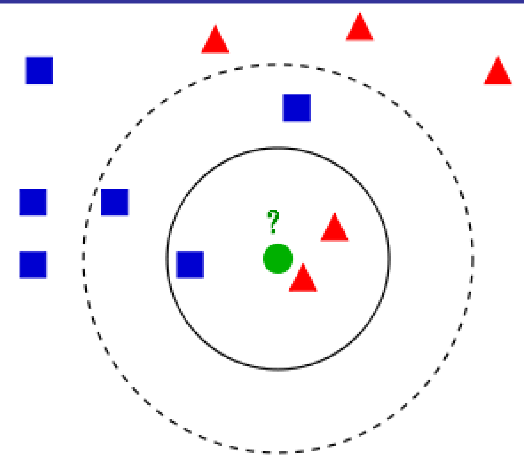
Input

- 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

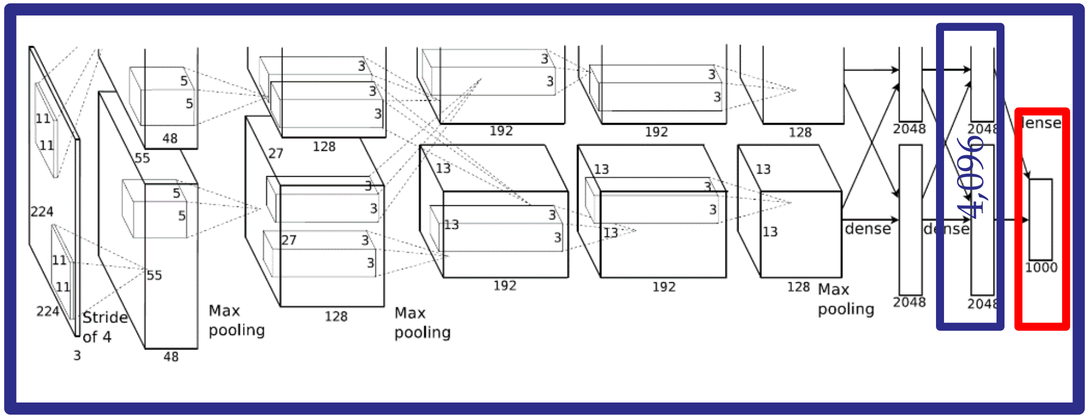
- *k*-nearest-neighbor (*k*NN) classifiers
  - Require an effective similarity model (features + distance function)
  - Search for the *k* most similar actions with respect to the query
  - Rank the retrieved actions to estimate the query class (probability)
- Machine-learning (ML) classifiers
  - Learn 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

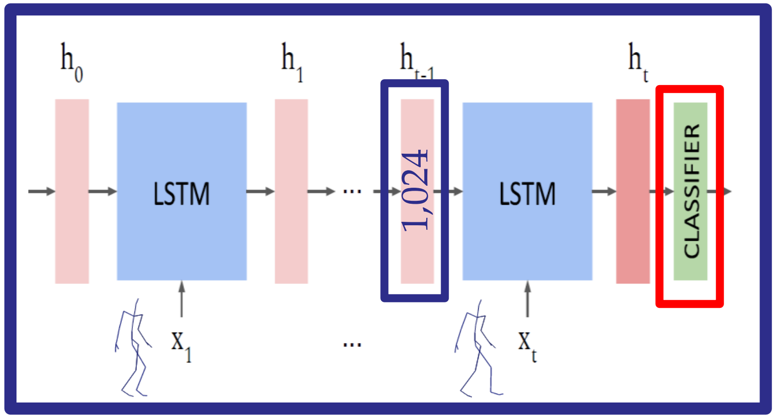
6.1 ML-Based Classification

Neural network classifiers

- Suitable architectures:
  - CNN or LSTM neural networks
- Training a network with labelled (categorized) actions
  - (Re)Training is time-consuming
- Classification only into known classes provided within the training process

4kD CNN **1kD LSTM**





6.2 1NN Classification

1NN classification

- Searching for the nearest neighbor based on similarity of features
- 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 KICK**
2. 9.1 JUMP
3. 10.2 JUMP
4. 14.3 KICK

☹ ⇒ **KICK (100%)**

KICK class
feature vectors

<…, 8.93, 10.1, 2.43, …>

<…, 7.42, 7.14, 2.27, …>

<…, 3.93, 6.26, 3.41, …>

6.2 *k*NN 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 relevant class samples
  - Similarities of neighbors are not considered
  - Example: query action of the **jump** class

k=4:
1. 8.7 KICK
2. 9.1 JUMP
3. 10.2 JUMP
4. 14.3 KICK

⇒ JUMP (50%)
⇒ KICK (50%)

6.2 *k*NN Classification

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 KICK	1. 0.55 KICK	1. 0.45 KICK	0.54 KICK ⇒ KICK (41%)
2. 9.1 JUMP	2. 0.58 JUMP	2. 0.42 JUMP	0.77 JUMP ⇒ **JUMP (59%)**
3. 10.2 JUMP	3. 0.65 JUMP	3. 0.35 JUMP	
4. 14.3 KICK	4. 0.91 KICK	4. 0.09 KICK	

☺

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

      ⋮

6.2 Comparison of Classification Methods

ACM ICMR 2019 Ottawa

- 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

	Accuracy (%)	
	HDM-122	HDM-130
LieNet-2Blocks (Huang et al., CVPR 2017)	N/A	75.78
CNN on motion images (Laraba et al., CAVW 2017)	N/A	83.33
Multi-scale filtering version of STGC (Li et al., AAAI 2018)	N/A	86.17
4kD CNN motion-image features + 1NN (Sed., MTaP 2018)	87.24	86.79
4kD CNN & handcrafted features + *k*NN (Sed., DEXA 2018)	89.09	88.78
1kD LSTM features + 1NN (2019)	91.41	90.74
1kD LSTM features + augmented training data + *k*NN (2019)	94.33	93.64

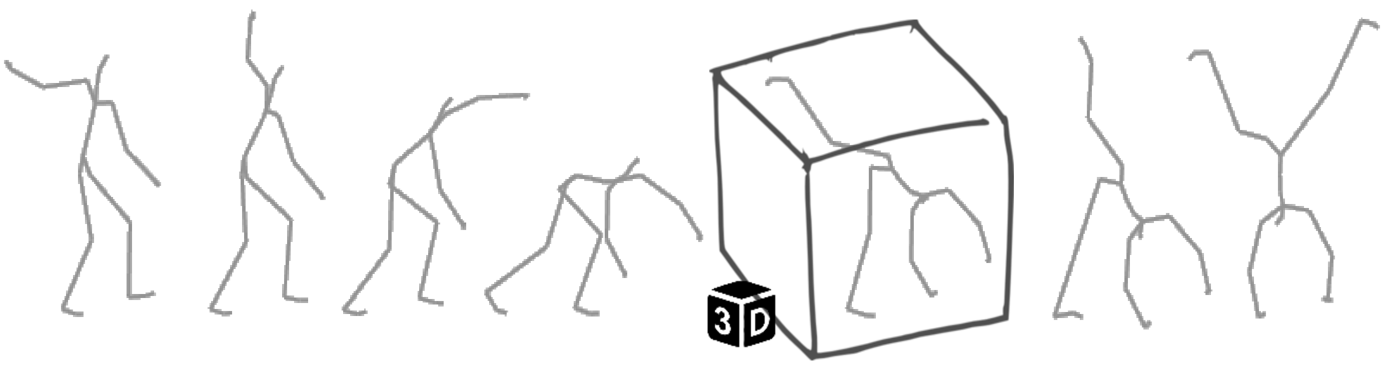
Advantages/disadvantages of the *k*NN and ML classifiers

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

Live Demo: http://disa.fi.muni.cz/mocap-action-recognition/

[Sedmidubsky et al.: Recognizing User-Defined Subsequences in Human Motion Data. ICMR, 2019]

7 Indexing and Searching in Long Motion Sequences



7.1 Long Motions

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 manner

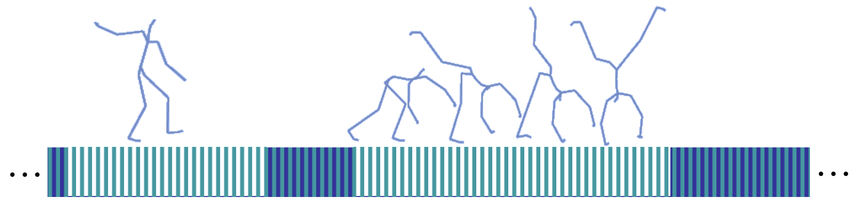
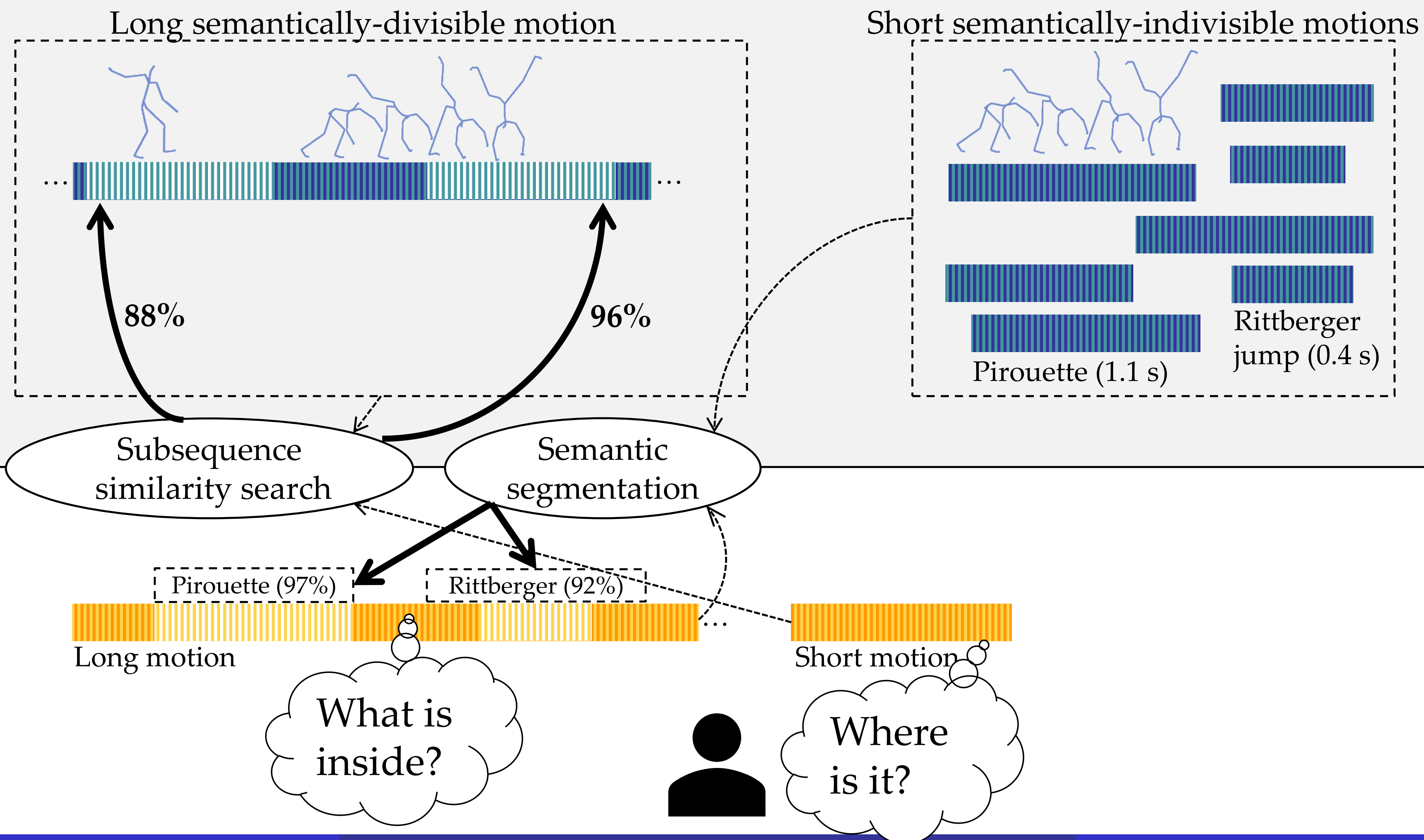


Figure skating performance (3 mins)

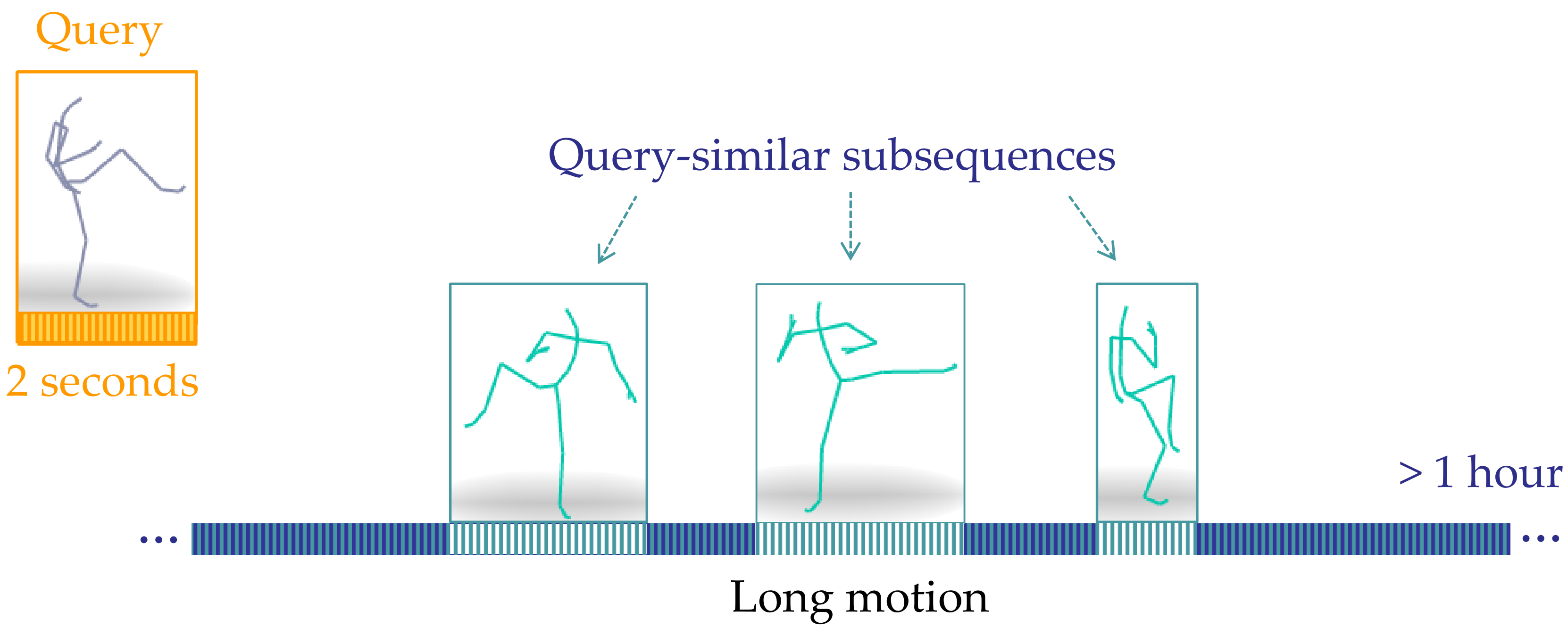
7.1 Processing Long Motions



7.2 Subsequence Search

Subsequence search

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



7.2 Search Challenges

ACM ICMR 2019 Ottawa

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

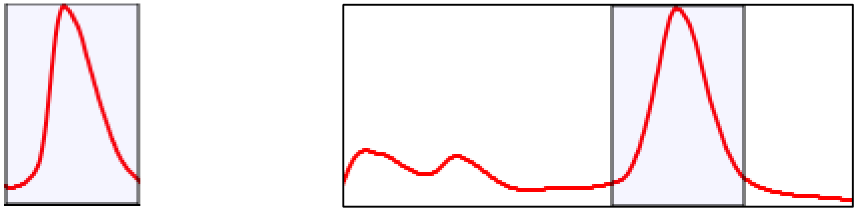
- Motion data can be perceived as a set of synchronized time series ~ a single multi-dimensional time series
  - E.g., a single time series for each joint and axis ($x/y/z$)

    => 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]



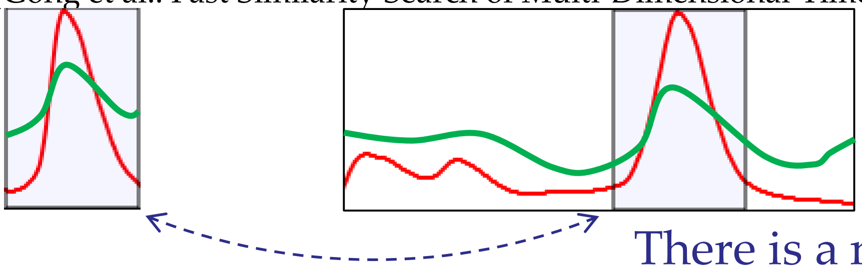
ACM ICMR 2019 Ottawa

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

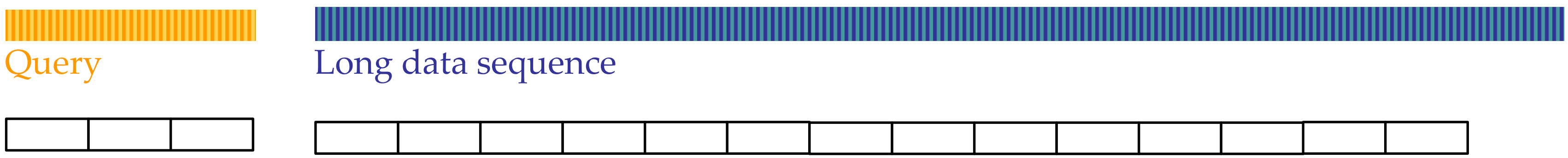
- Traditional time-series algorithms hardly applicable to motion-data domain due to the absence of distance functions working **effectively** on **frame-based features**

7.2 Subsequence Search in Motion Data

ACM ICMR 2019 Ottawa

Subsequence matching in motion data

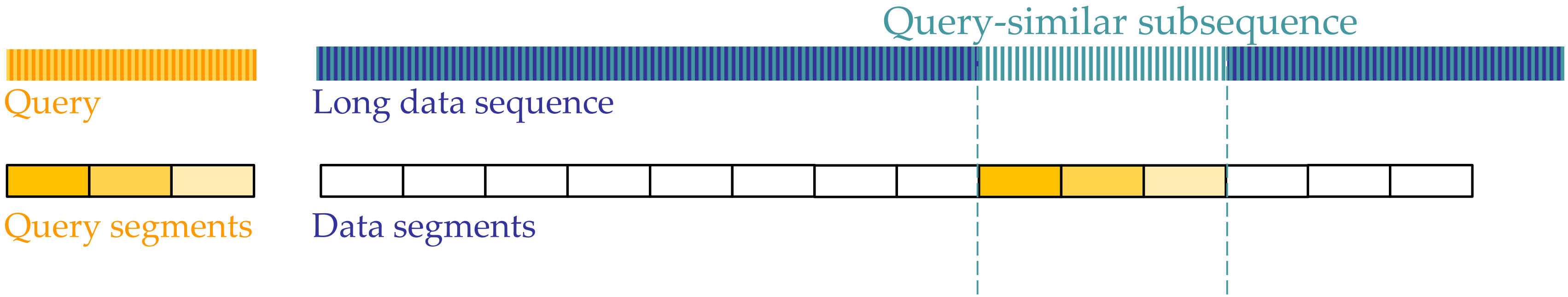
- Effective motion-based features are extracted from short motions => segmentation
- Partitioning the query and long motion sequence into parts – segments – to be meaningfully comparable



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

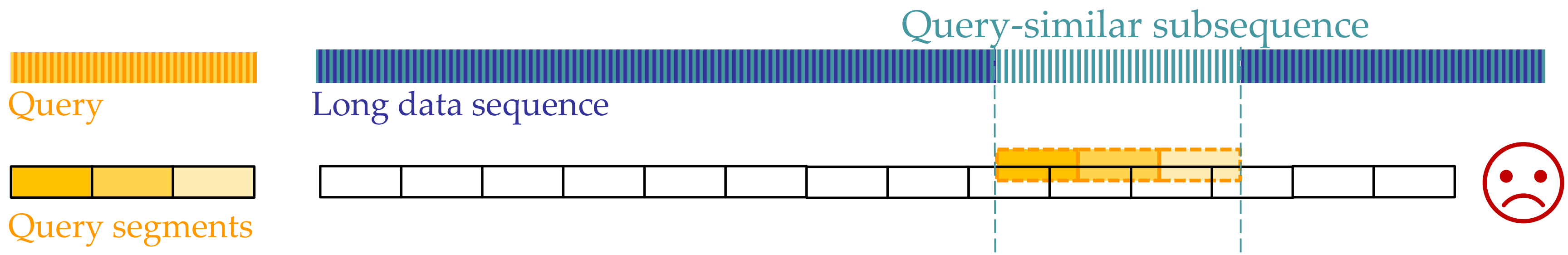
Subsequence matching in motion data

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



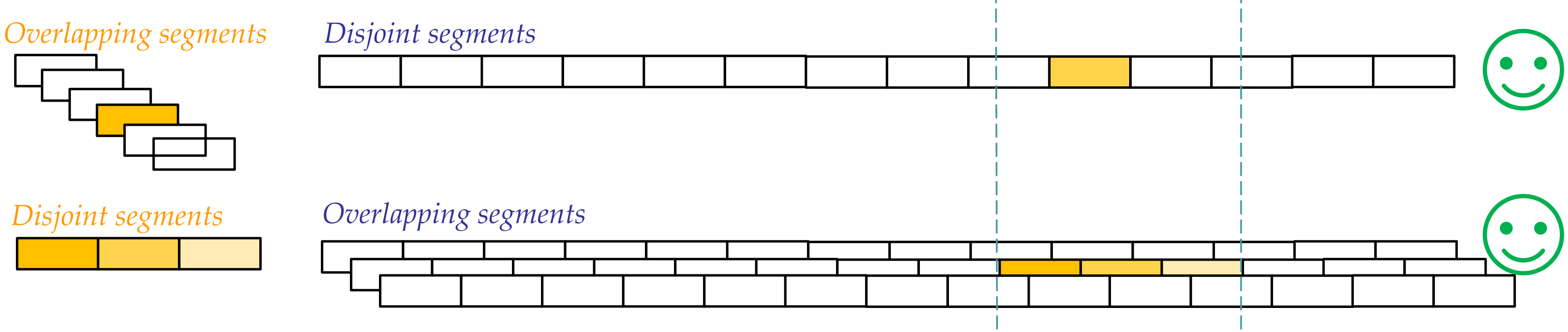
Alignment problem in subsequence matching

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



⇒ Solving the alignment problem by overlapping segments

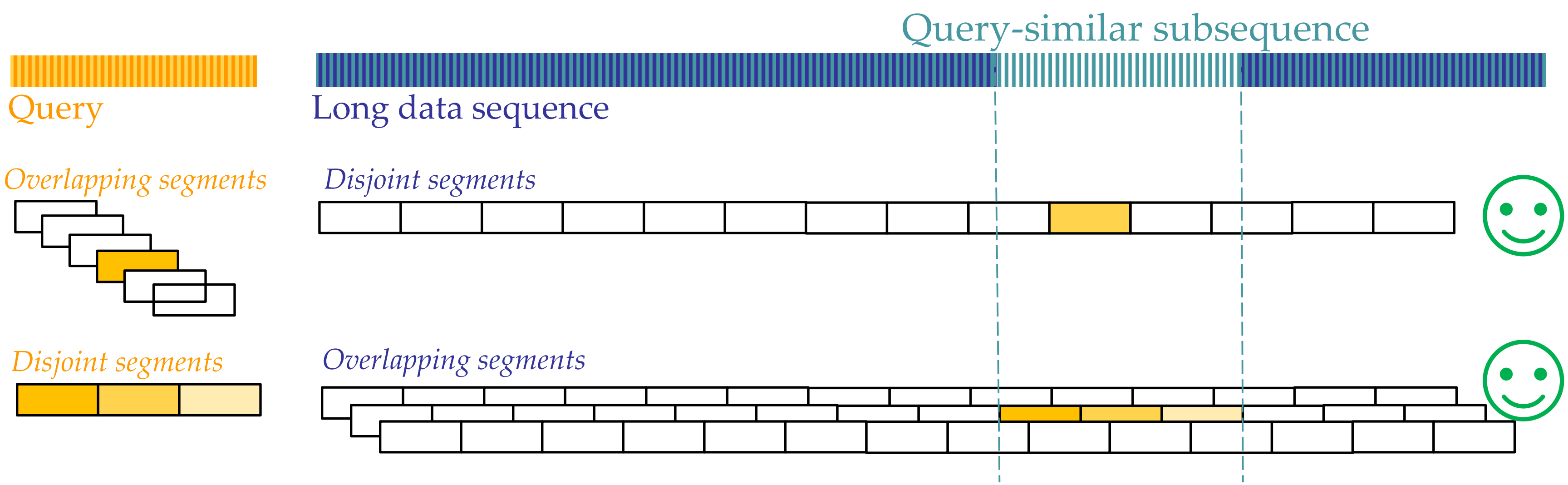
– Considering every possible segment is extremely expensive



7.2 Overlapping Segmentation

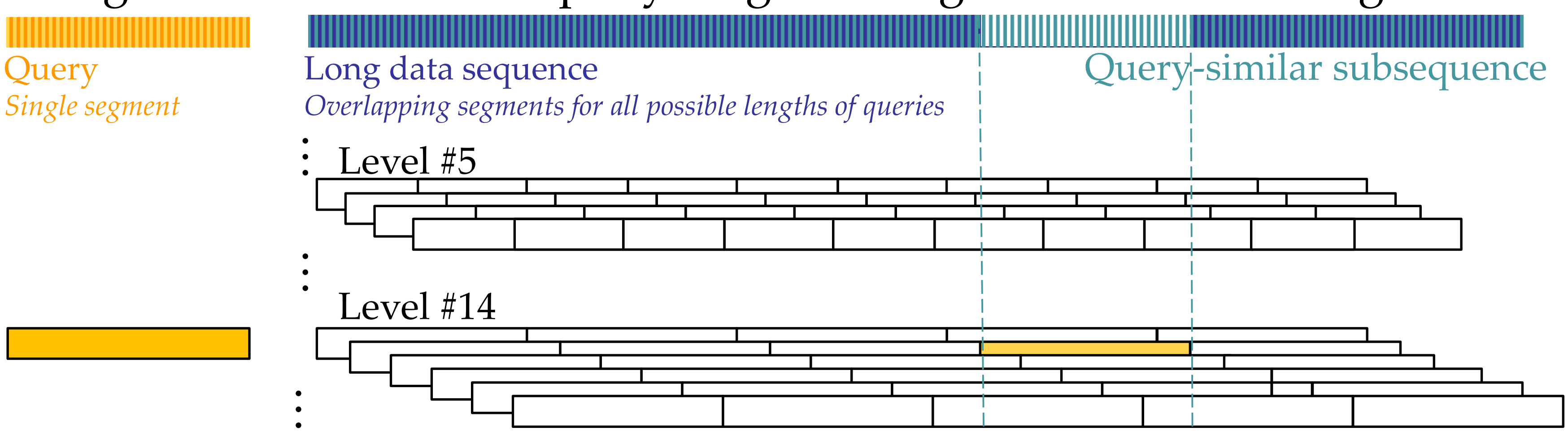
Partitioning both the query and data sequence

- ☺ 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

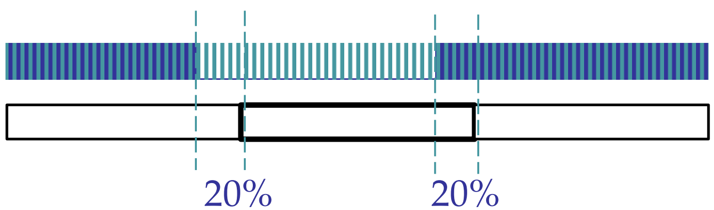


[Sedmidubsky et al.: Similarity Searching in Long Sequences of Motion Capture Data. SISAP, 2016]

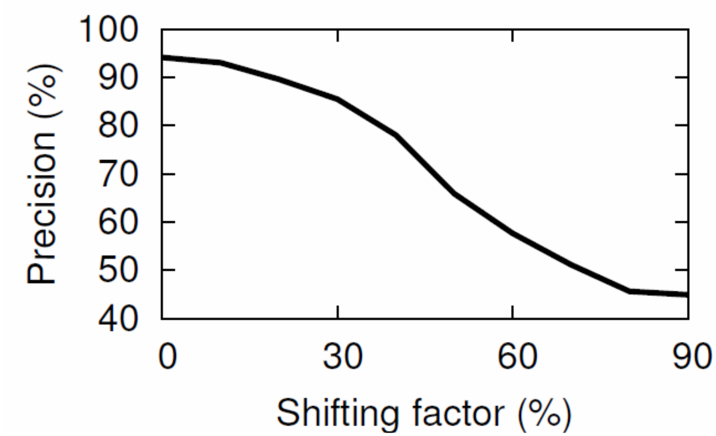
7.2 Elasticity Property

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)

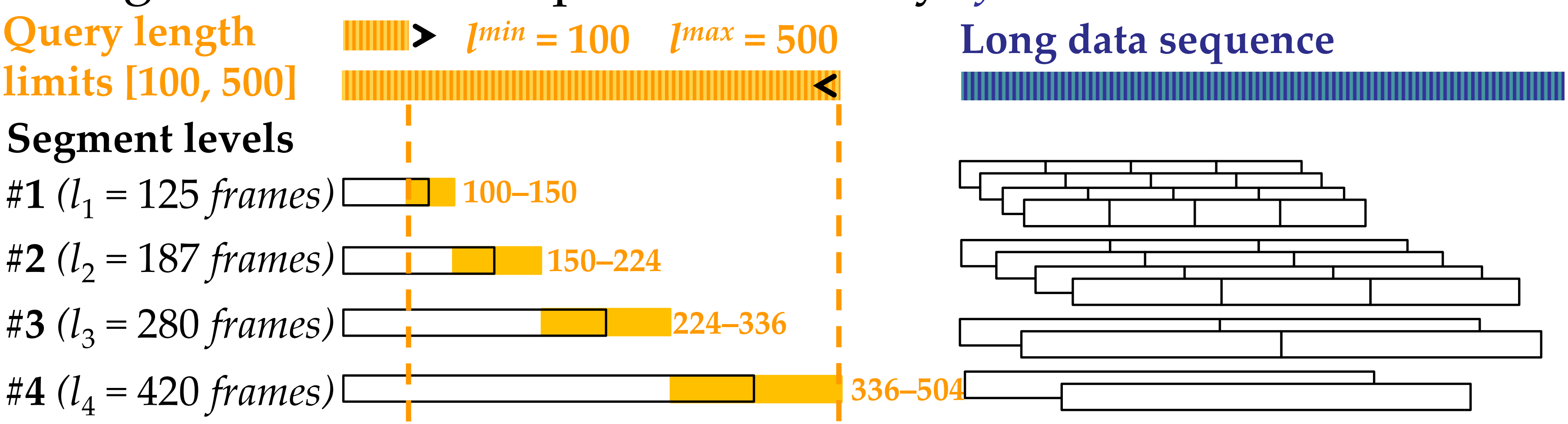
Levels can be generated only for the specific lengths of queries (and not for all the possible ones)

☺ The big number of segments can be dramatically reduced

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 $cf = 0.2 \sim 20\%$:

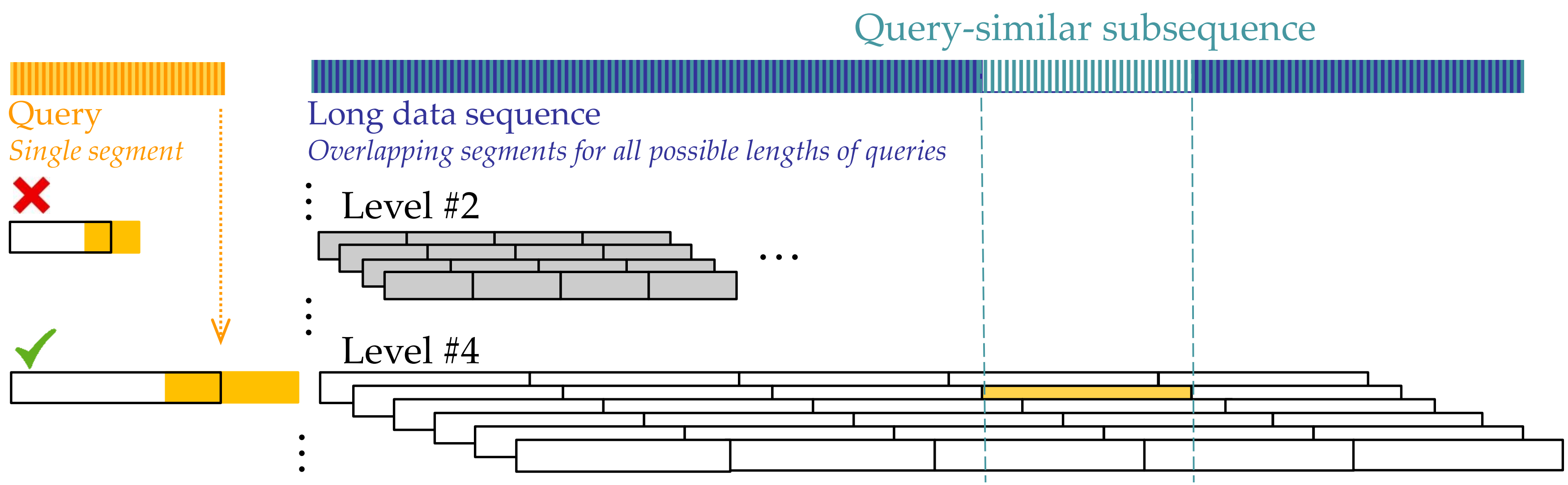


Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

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



7.2 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

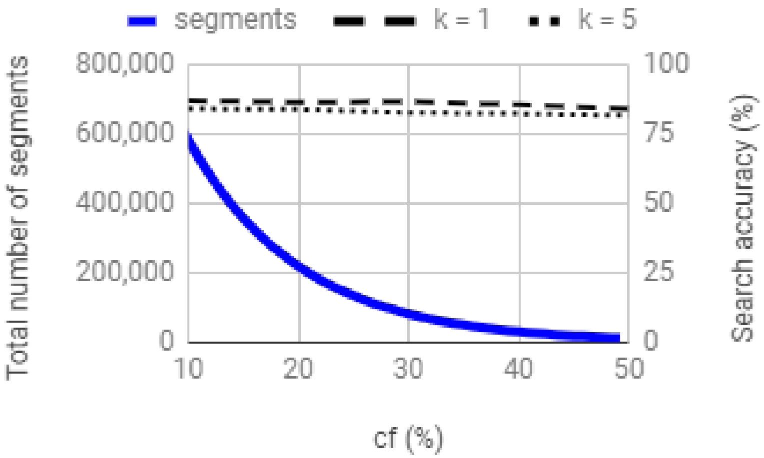
HDM05 – long motions

- 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%}

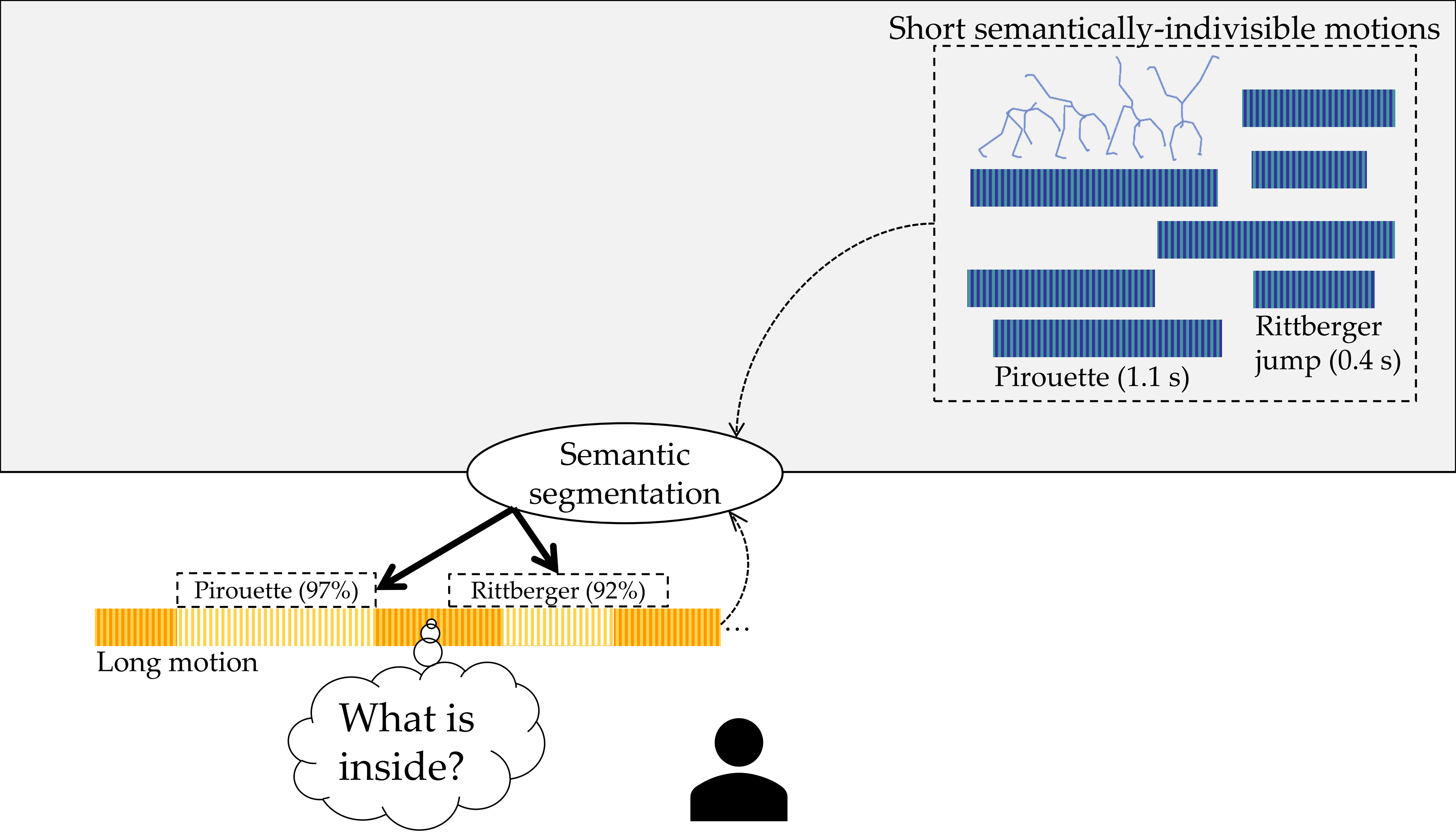


cf [%]	# of levels	Sequential scan [ms]
10	18	447
20	9	205
30	6	126
40	5	88
50	4	66

Summary

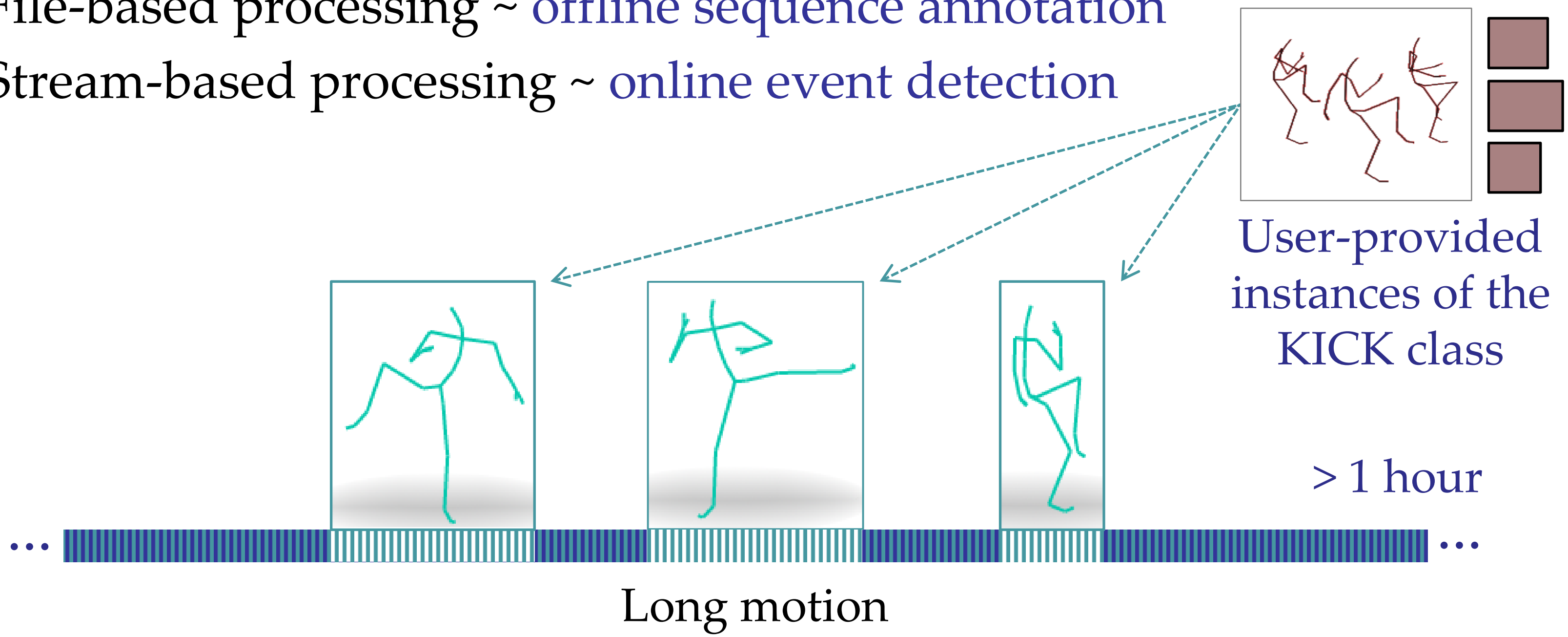
- Advanced subsequence matching in mocap data:
  - Query always considered as a single segment
  - The elasticity property of the motion-image similarity concept dramatically reduces the number of data segments
- Efficiency:
  - Searching the 68-minute sequence sequentially takes 205ms
  - Search times 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
- Live demo: http://disa.fi.muni.cz/mocap-demo/

7.3 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



7.3 Semantic Segmentation

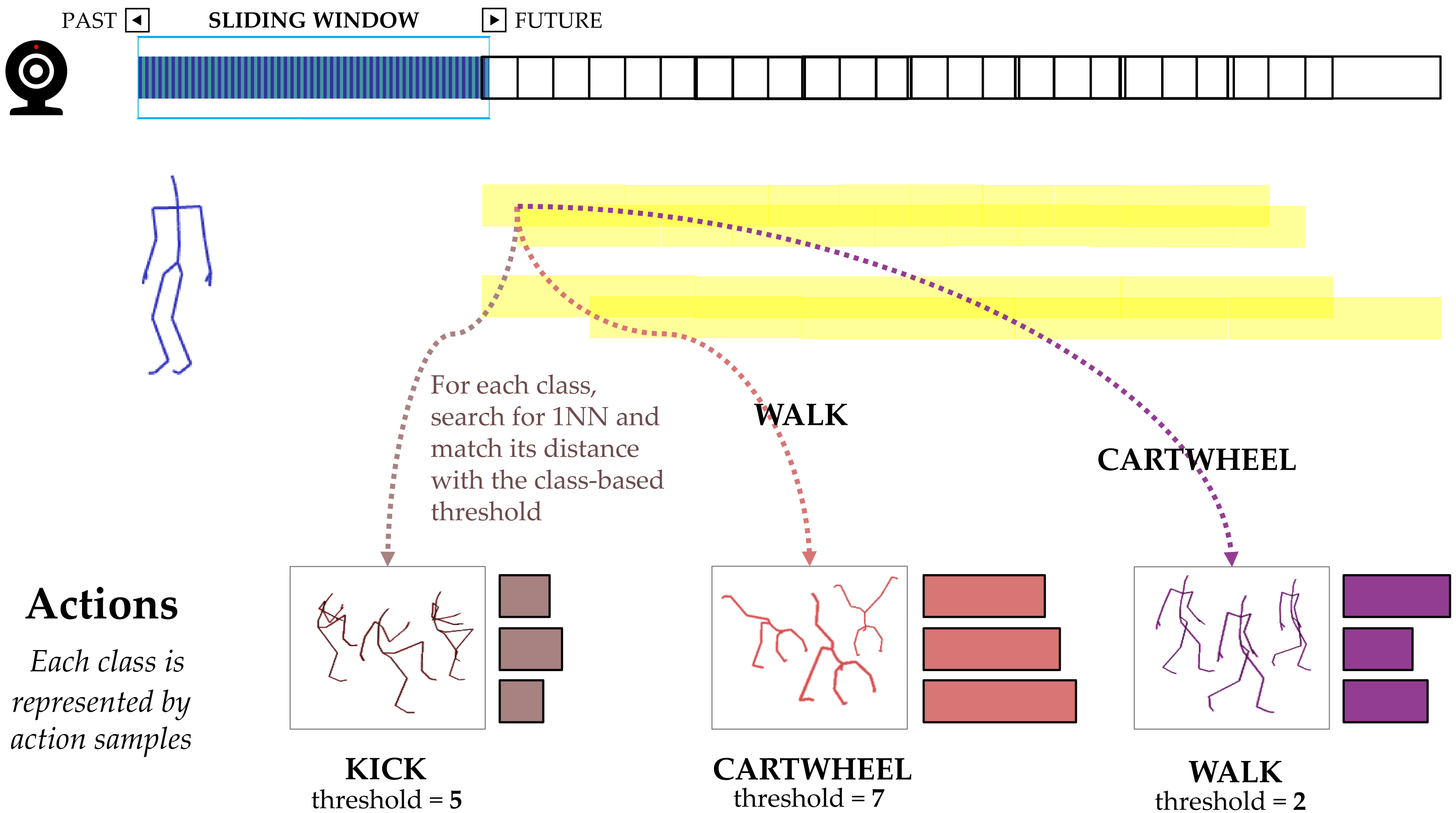
Challenges

- 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
- [Carrara et al.: LSTM-Based Real-Time Action Detection and Prediction in Human Motion Streams. Multimedia Tools and Applications, 2019]
  - Offline-LSTM – offline sequence annotation
  - Online-LSTM – online event detection

7.3 Segment-Based Event Detection

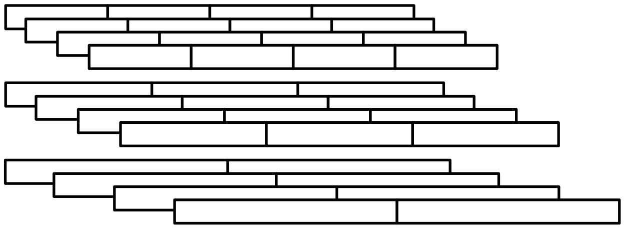


Actions

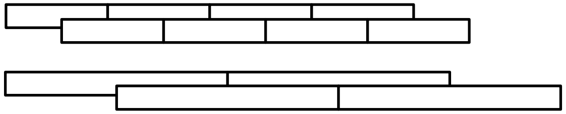
Each class is represented by action samples

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.

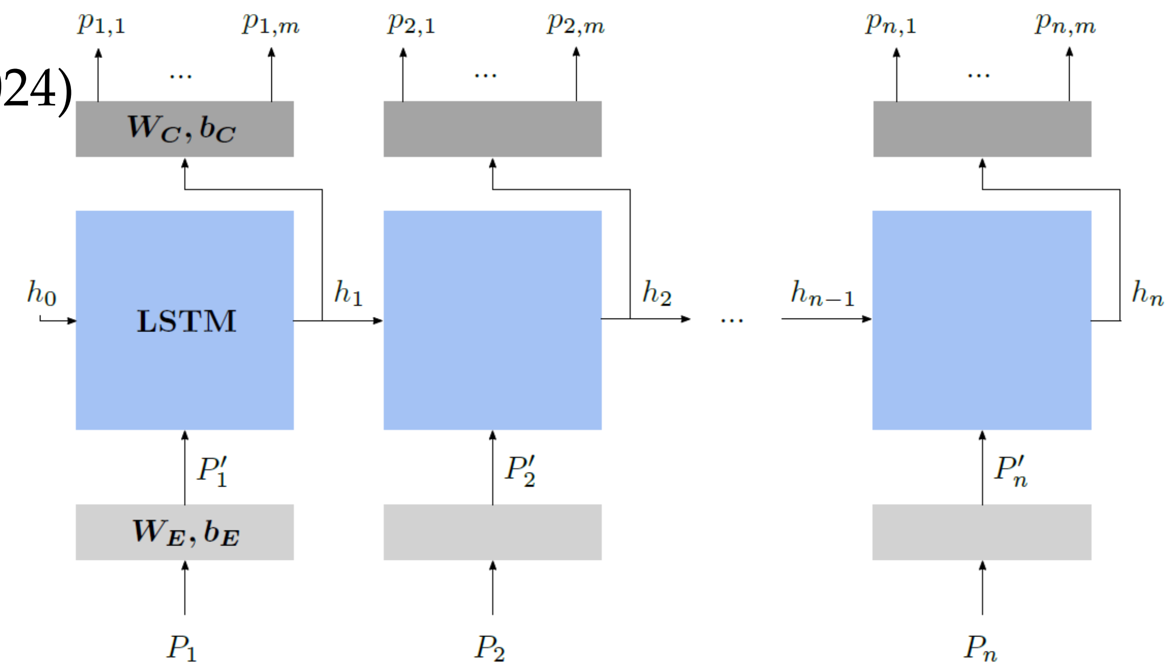


Sparse segmentation
Produces less segments but requires a more elastic similarity measure.

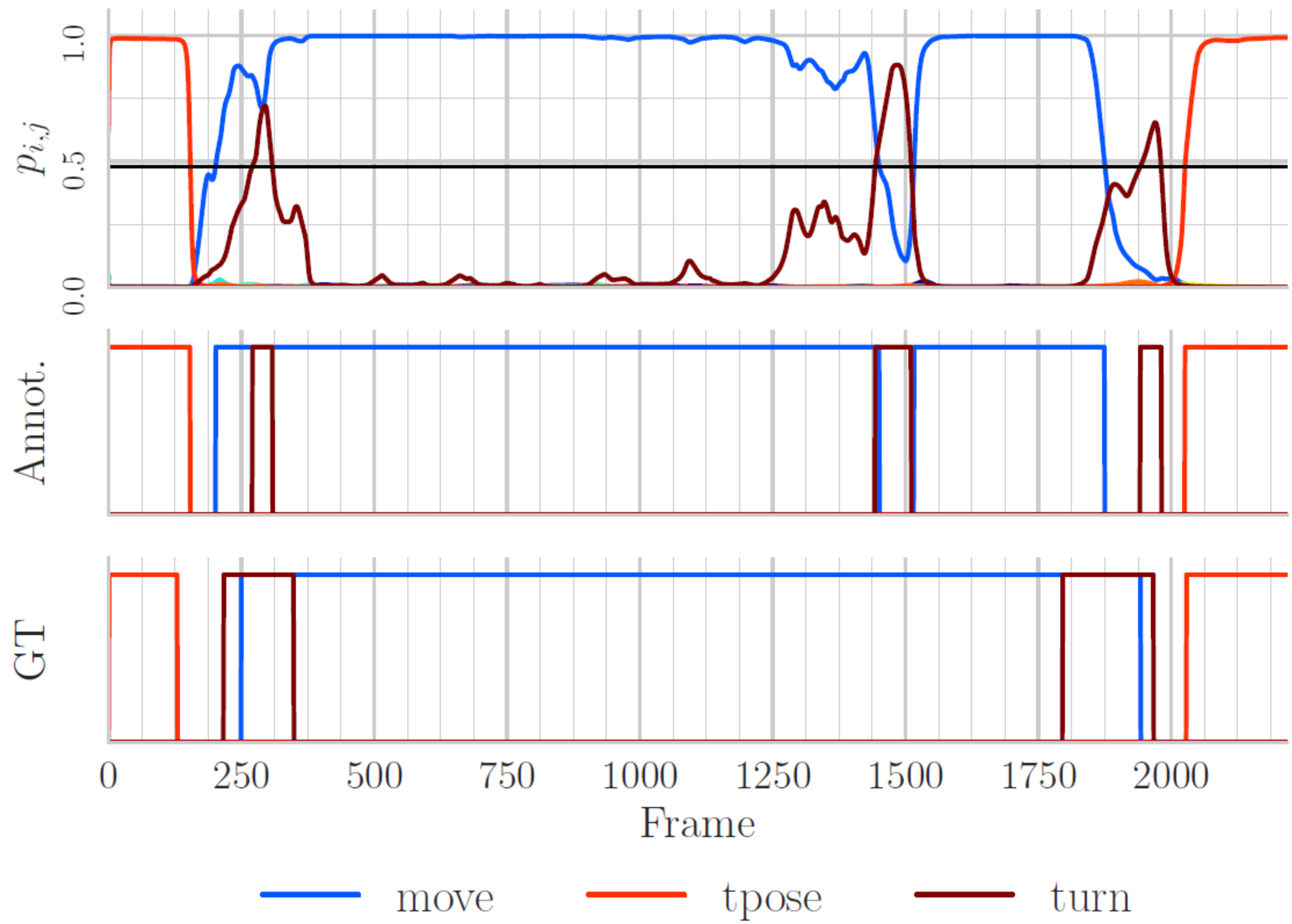
- Segmentation density impacts efficiency and effectiveness

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

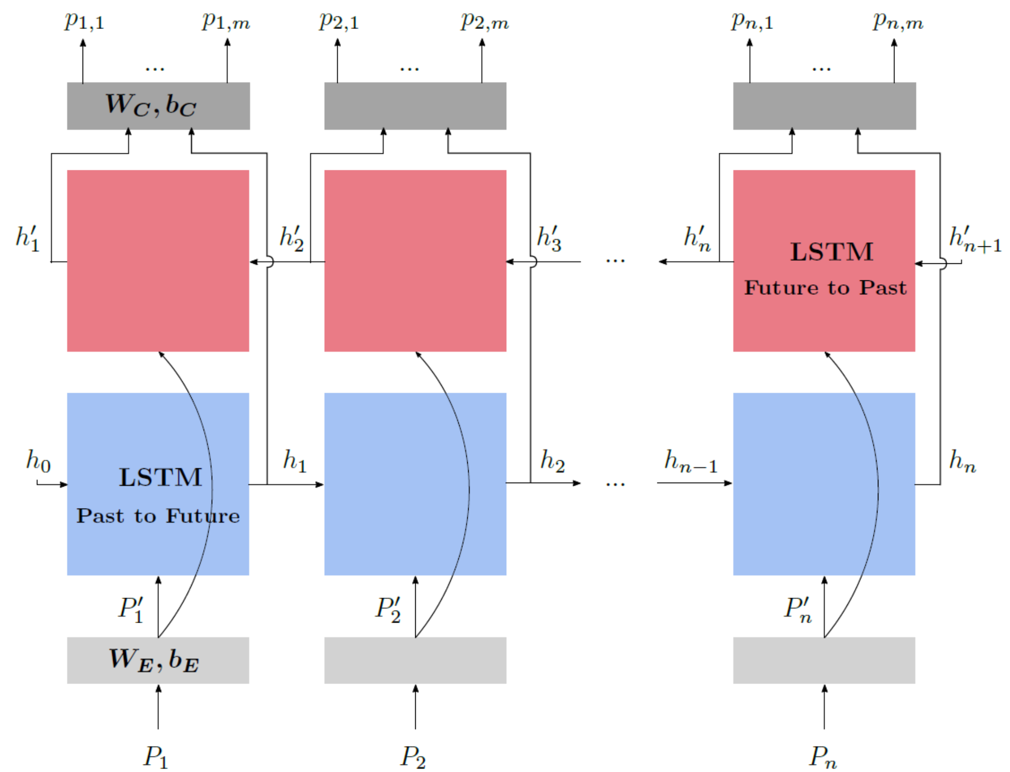


Output of Online-LSTM



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

7.3 Comparison of Methods

Accuracy measure

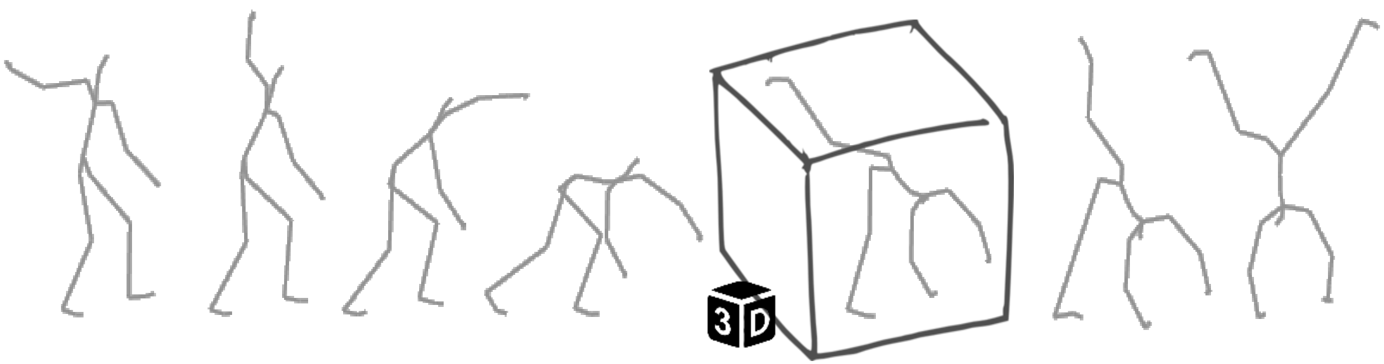
- F_1 score – a harmonic mean of recall and precision measured on the level of individual frames

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

  - 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 (2019)	17 min	51 min	5 h	-	0.1 ms	0.1 ms	74.95 %
Offline-LSTM (2019)	17 min	51 min	3.5 h	-	0.1 ms	0.1 ms	78.78 %

8 Conclusions



8 Conclusions

Tutorial objectives:

- To present challenges and existing principles for searching in mocap capture data
  - **Presented operations** – similarity comparison, subsequence search, action recognition, 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

Subsequence search demo

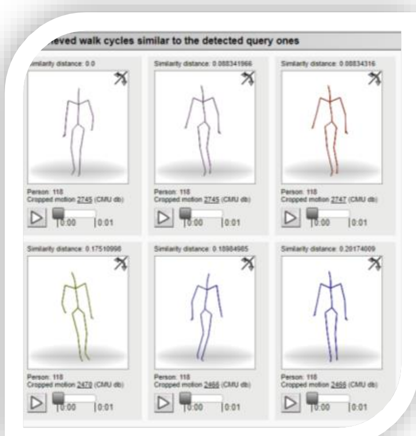
- http://disa.fi.muni.cz/mocap-demo/

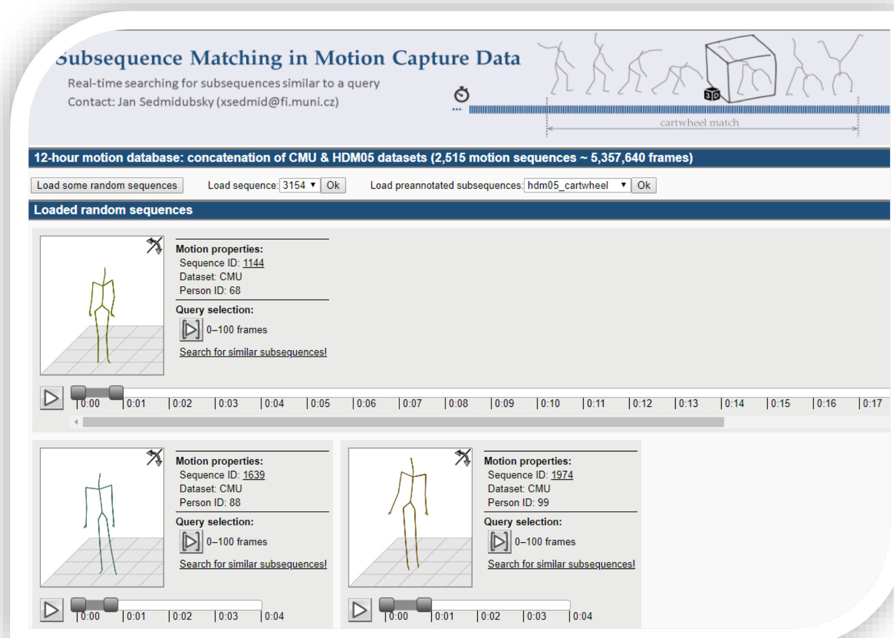
Action recognition demo

- http://disa.fi.muni.cz/mocap-action-recognition/

Gait recognition demo

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





8 DISA at Masaryk University

Laboratory of
Data Intensive Systems and Applications

http://disa.fi.muni.cz

8 SISAP Conference

ACM ICMR 2019 Ottawa

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
2019 Newark NJ USA

2008 Cancun Mexico
2010 Instanbul Turkey
2012 Toronto Canada
2014 Los Cabos Mexico
2016 Tokyo Japan
2018 Lima Peru

Resources

ACM ICMR 2019 Ottawa

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.]

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.]
- [Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. Deep Motifs and Motion Signatures. ACM Transactions on Graphics, 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.]
- [Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. Deep Motifs and Motion Signatures. ACM Transactions on Graphics, 2018.]

Resources

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 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.]
- [Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep Learning on Lie Groups for Skeleton-Based Action Recognition. CVPR, 2017.]
- [Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Ahmed Sohel, and Farid Boussaïd. A New Representation of Skeleton Sequences for 3D Action Recognition. CVPR, 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, 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.]

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 ICMR 2019 Ottawa

Presentations

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

Funding

- Supported by the Czech Science Foundation project No. GA19-02033S.