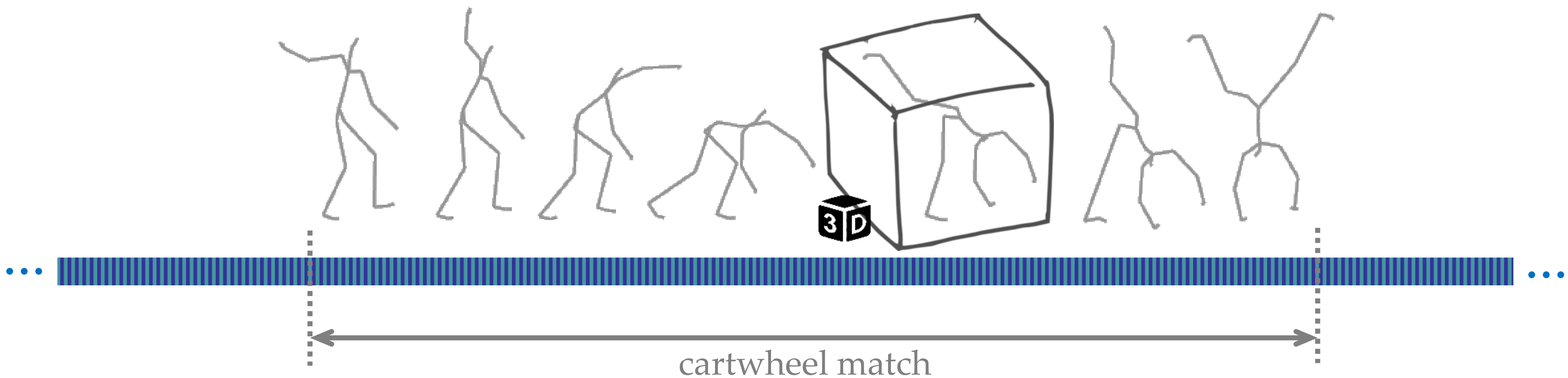
Similarity Searching in Long Sequences of Motion Capture Data

Jan Sedmidubsky, Petr Elias, Pavel Zezula



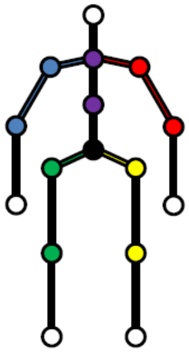
Laboratory of Data Intensive Systems and Applications
disa.fi.muni.cz

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

Introduction to Motion Capture Data

Motion Capture (mocap) Data

- Acquired by marker-based/less capturing technologies
- Complex multi-dimensional spatio-temporal data
- 3D space, 25+ body joints, 30+ frames per second
- Input for our research



Simplified human skeleton with 16 joints

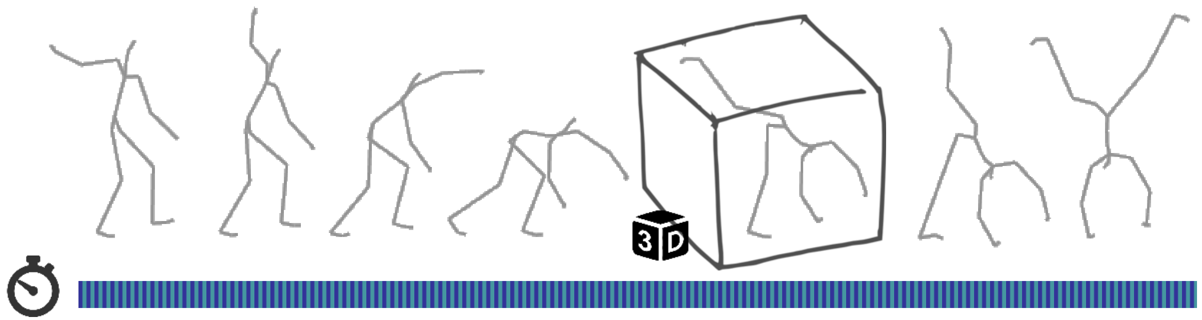


Illustration of short cartwheel motion sequence
5 seconds of 120 Hz mocap data represent 55,800 float numbers

Applications of Mocap Data

Computer Animation

Finding desired actions for a game or movie from a databank of motion recordings

Medicine

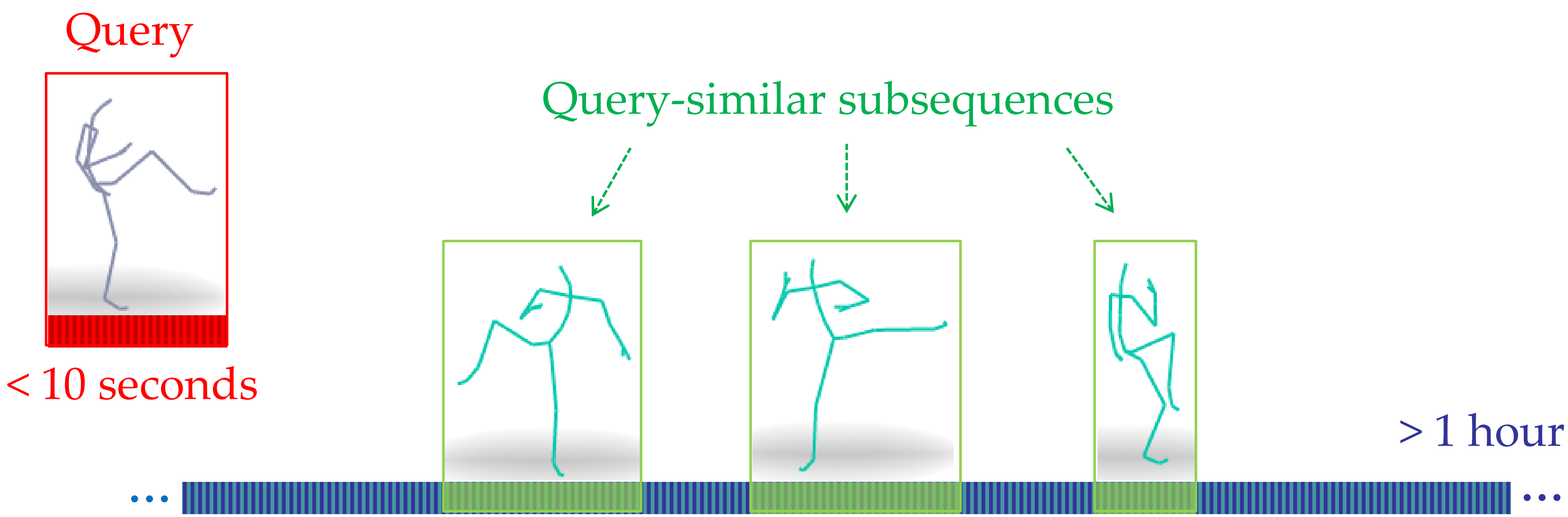
Recognizing developmental disabilities and movement disorders such as cerebral palsy

Sports

Searching for similar movement patterns to analyze athlete performance

Objective – Subsequence Matching

Objective – to develop an efficient mechanism for searching a long data sequence and localizing its parts that are similar to a short query sequence



Long data sequence

(can be a concatenation of multiple sequences)

Subsequence Matching – Challenges

- Actors have **different bodies** *(e.g., child and adult)*
- Seemingly **same actions** can be performed in **different speeds** *(faster, slower)* and **styles** *(e.g., frontal kick vs. side kick)*
- Captured data can be **noisy or incomplete**

Subsequence Matching – Challenges

- Actors have **different bodies** *(e.g., child and adult)*
- Seemingly **same actions** can be performed in **different speeds** *(faster, slower)* and **styles** *(e.g., frontal kick vs. side kick)*
- Captured data can be **noisy or incomplete**

#1 Robust Similarity Measure

Subsequence Matching – Challenges

- Actors have **different bodies** *(e.g., child and adult)*
- Seemingly **same actions** can be performed in **different speeds** *(faster, slower)* and **styles** *(e.g., frontal kick vs. side kick)*
- Captured data can be **noisy or incomplete**

#1 Robust Similarity Measure

- **Query** can potentially **occur anywhere** in a long sequence
- **Query** can be potentially **any short sequence** *(e.g., semantic action such as kick or jump, its part or a transition in between any of these)*

Subsequence Matching – Challenges

- Actors have **different bodies** *(e.g., child and adult)*
- Seemingly **same actions** can be performed in **different speeds** *(faster, slower)* and **styles** *(e.g., frontal kick vs. side kick)*
- Captured data can be **noisy or incomplete**

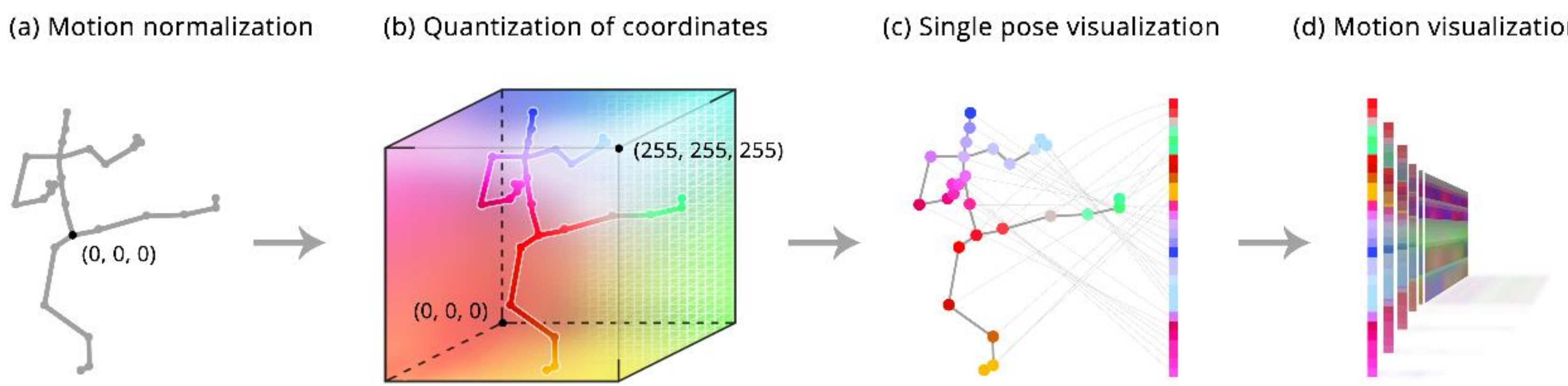
#1 Robust Similarity Measure

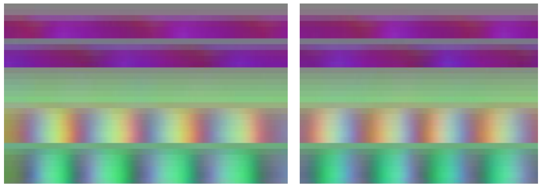
- **Query** can potentially **occur anywhere** in a long sequence
- **Query** can be potentially **any short sequence** *(e.g., semantic action such as kick or jump, its part or a transition in between any of these)*

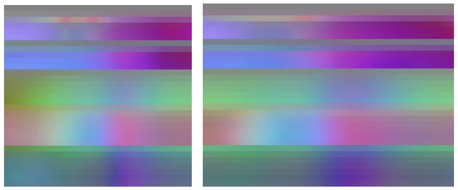
#2 Efficient Subsequence Matching

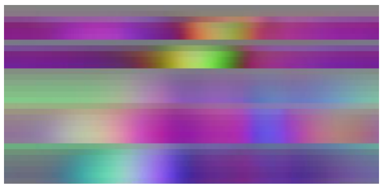
Our motion similarity – **4,096D** features + L_2 metric

- Mocap data are encoded into RGB images [Elias et al., SISAP 2015]
- Features extracted from RGB images using a deep convolutional neural network that performs very well on image data



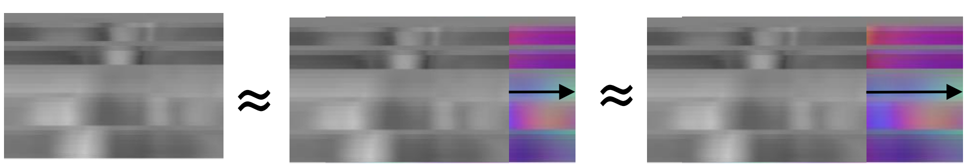

Rotate arms


Stand up

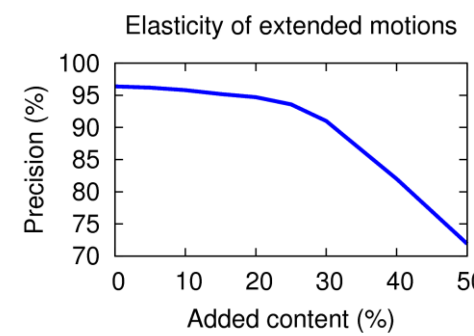

Cartwheel

#1 Similarity Measure – Properties

- **Efficiency**
  - Motions of **different lengths** have the **fixed-size** features
  - L_2 comparison enables a utilization of any **metric-based index**
- **Effectiveness**
  - Copes well with different speeds and styles of actions
  - **Elasticity** – similarity distances change only slightly when **content is removed or added** **(important for sequence segmentation)**



original **+ 10 % added** **+ 20 % added**



Sensitivity to an added/removed content
Adding a bounded amount of extra content has a minor effect to the search precision. A similar trend can be observed when a similar amount of content is removed.

Subsequence Matching – Challenges

- Actors have **different bodies** *(e.g., child and adult)*
- Seemingly **same actions** can be performed in **different speeds** *(faster, slower)* and **styles** *(e.g., frontal kick vs. side kick)*
- Captured data can be **noisy or incomplete**

✓ **#1 Robust Similarity Measure**

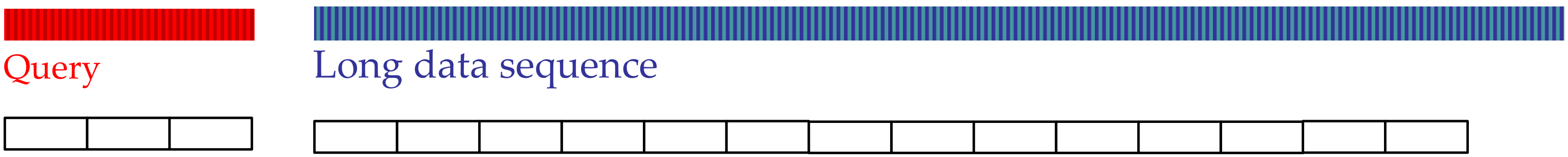
- **Query** can potentially **occur anywhere** in a long sequence
- **Query** can be potentially **any short sequence** *(e.g., semantic action such as kick or jump, its part or a transition in between any of these)*

→ **#2 Efficient Subsequence Matching**

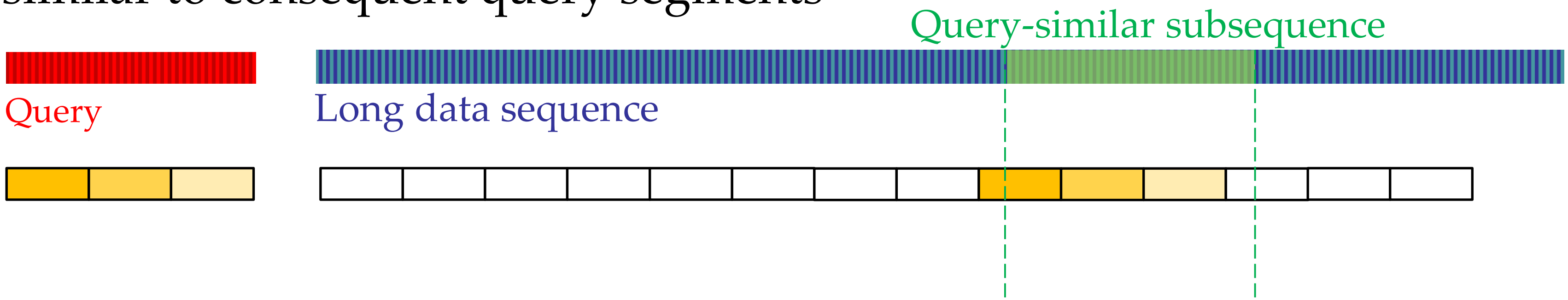
#2 Subsequence Matching

Subsequence matching:

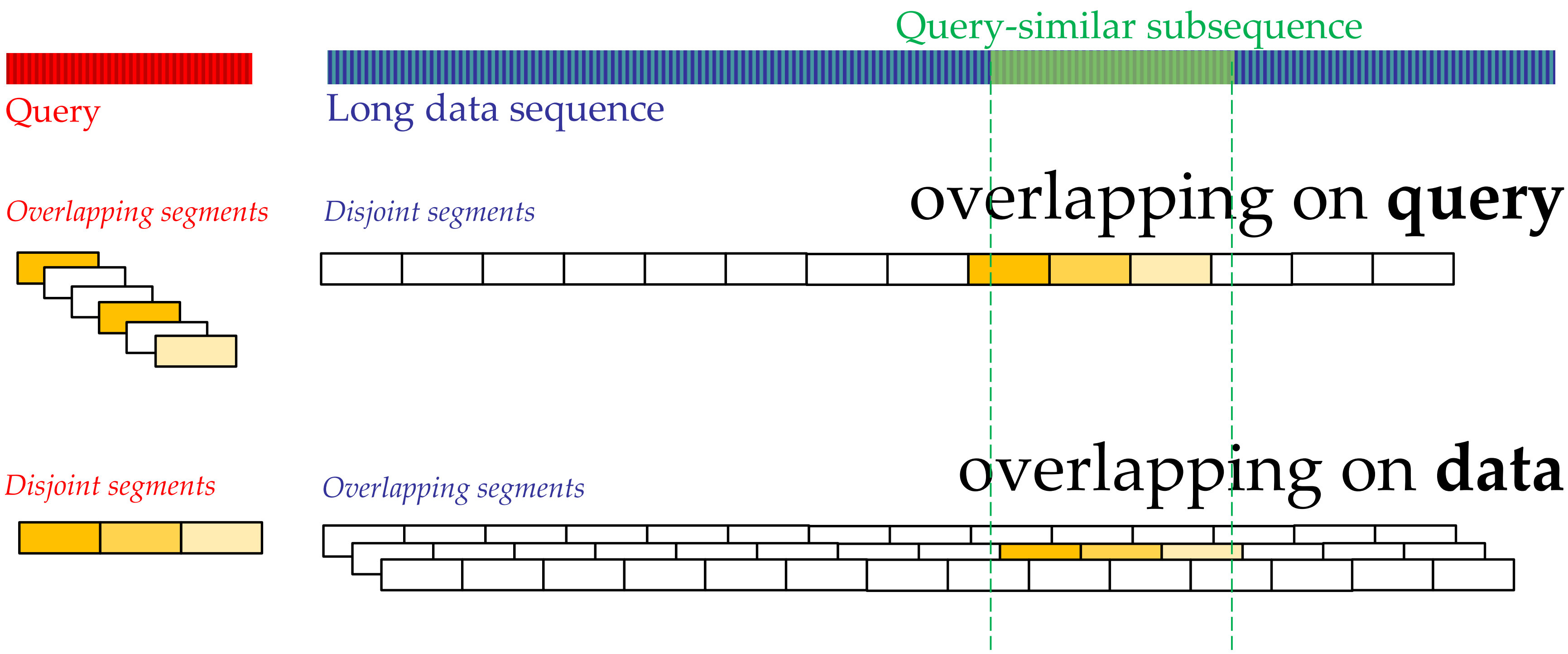
- **Segmentation** – short query and long data sequence are partitioned into parts (segments) to be meaningfully comparable (to have similar lengths)



- **Retrieval algorithm** – searching for consequent data segments that are similar to consequent query segments



#2 Segmentation – Overlapping on Query/Data

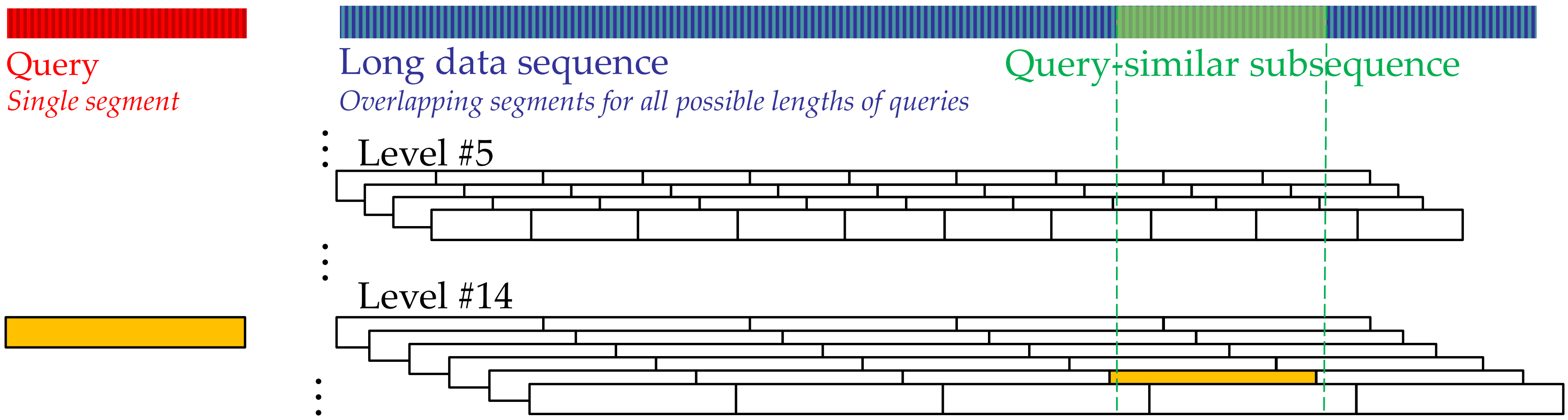


- ☹ A lot of query segments – longer queries are more expensive to evaluate
- ☹ Grouping relevant segments w.r.t. temporal information

#2 Segmentation – Naive

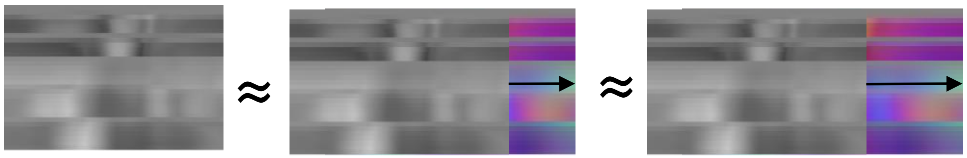
Query as a single segment – naive solution

- Query always considered as a **single segment**
- Data sequence as **multi-level overlapping segments**

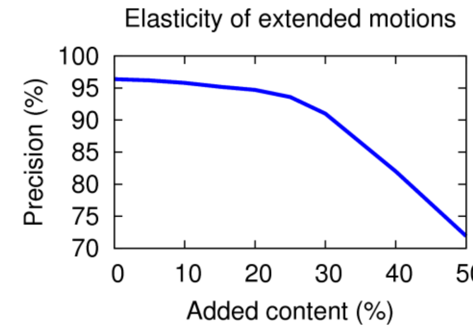


☺ Much easier retrieval – one query, no complex post-processing
☹ Segment level for each query length – a huge number of data segments

#1 Similarity Measure – Elasticity Property



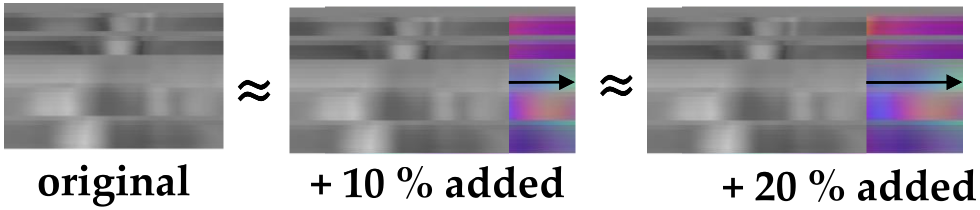
original **+ 10 % added** **+ 20 % added**

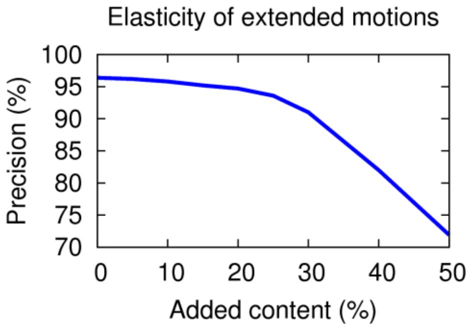


Sensitivity to an added/removed content

Adding a bounded amount of extra content has a minor effect to the search precision. A similar trend can be observed when a similar amount of content is removed.

#1 Similarity Measure – Elasticity Property



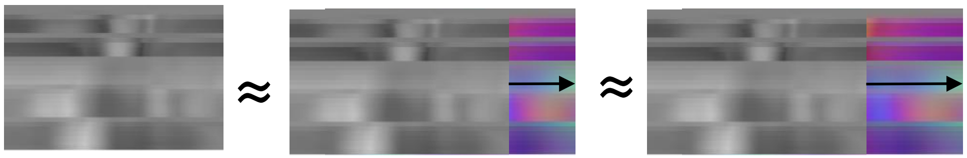


Sensitivity to an added/removed content

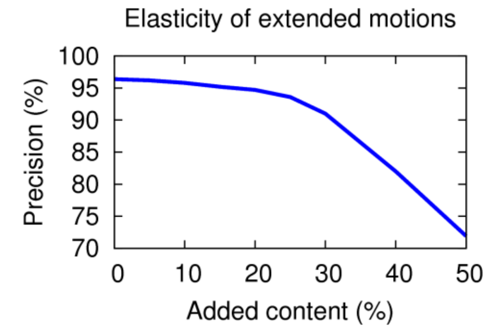
Adding a bounded amount of extra content has a minor effect to the search precision. A similar trend can be observed when a similar amount of content is removed.

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

#1 Similarity Measure – Elasticity Property



original ≈ **+ 10 % added** ≈ **+ 20 % added**



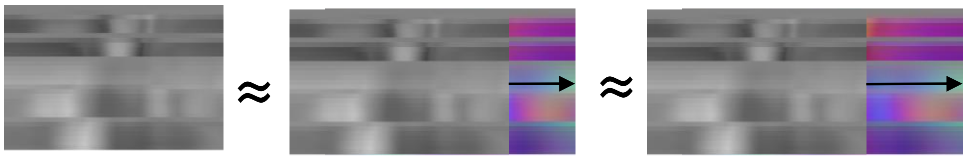
Sensitivity to an added/removed content

Adding a bounded amount of extra content has a minor effect to the search precision. A similar trend can be observed when a similar amount of content is removed.

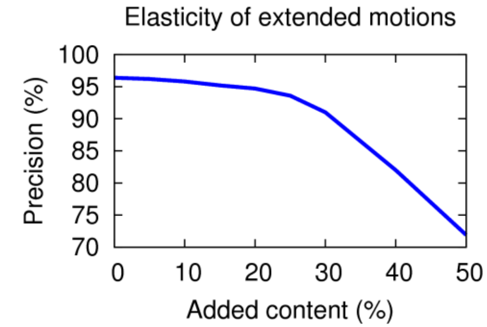
Overlapping segments can be shifted by 5–25 % of their length (and not only by a single frame)

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

#1 Similarity Measure – Elasticity Property



original ≈ **+ 10 % added** ≈ **+ 20 % added**



Sensitivity to an added/removed content

Adding a bounded amount of extra content has a minor effect to the search precision. A similar trend can be observed when a similar amount of content is removed.

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

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

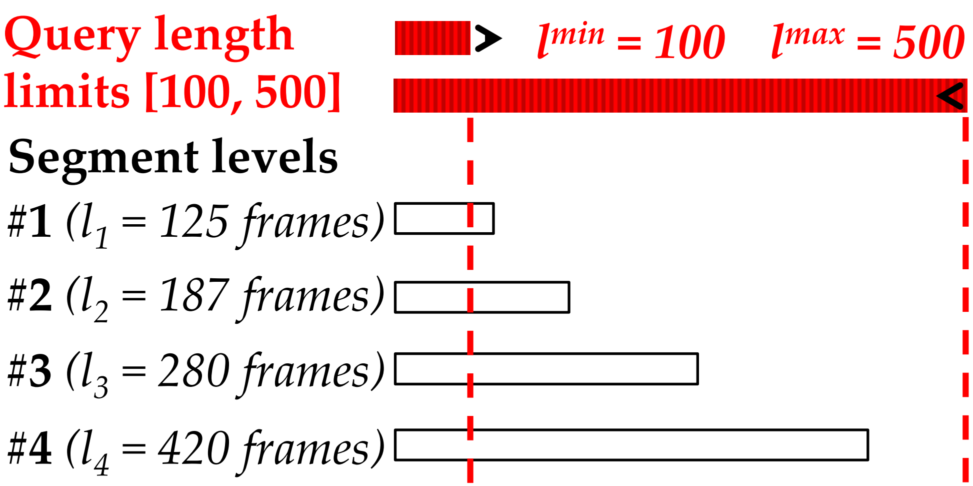
☺ The huge number of segments can be dramatically reduced!

#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:



#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:

Query length limits [100, 500]

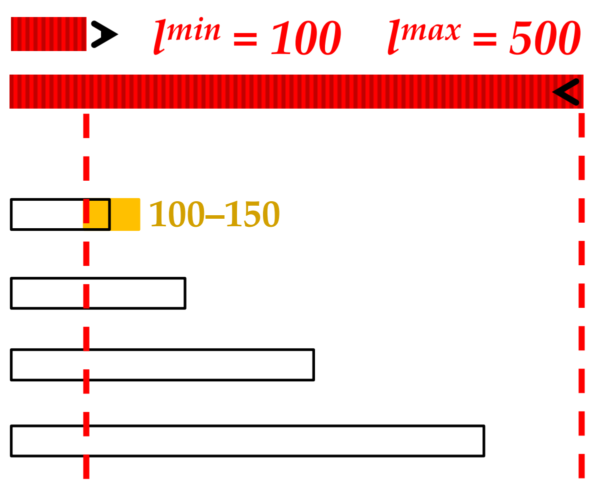
Segment levels

#1 (l_1 = 125 *frames*)

#2 (l_2 = 187 *frames*)

#3 (l_3 = 280 *frames*)

#4 (l_4 = 420 *frames*)



#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:

Query length limits [100, 500]

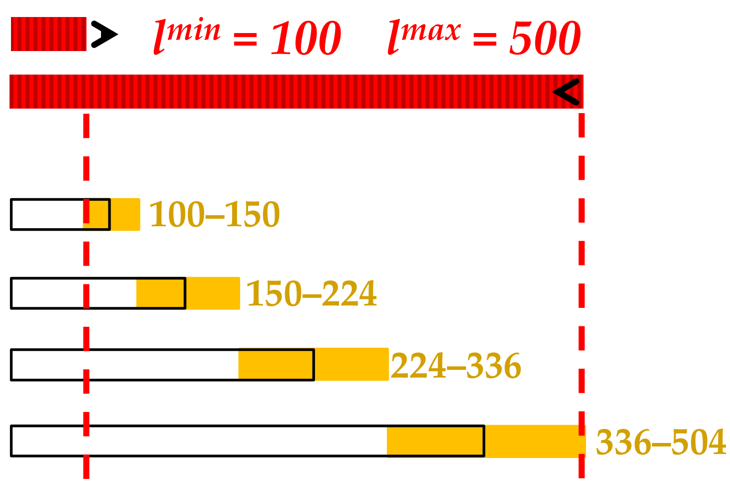
Segment levels

#1 (l_1 = 125 frames)

#2 (l_2 = 187 frames)

#3 (l_3 = 280 frames)

#4 (l_4 = 420 frames)

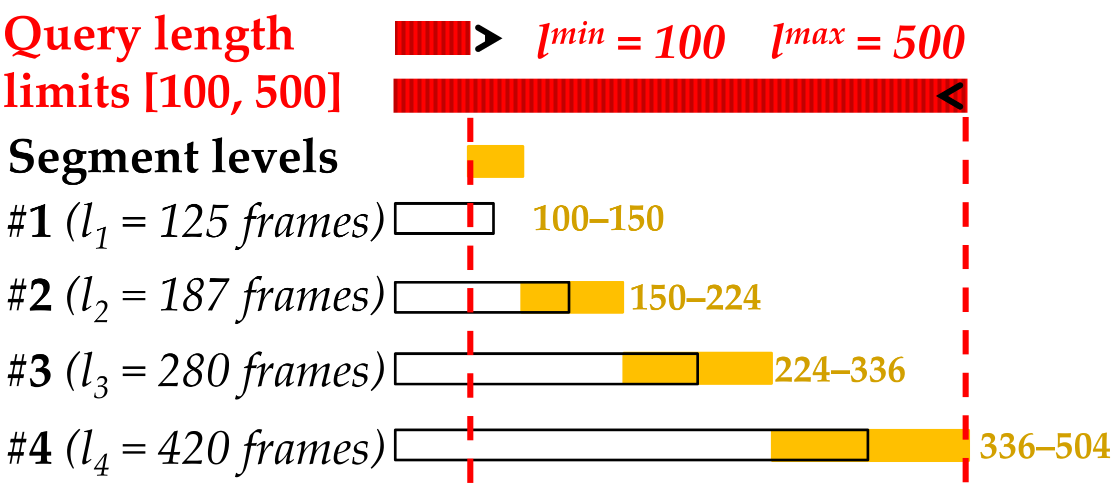


#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:

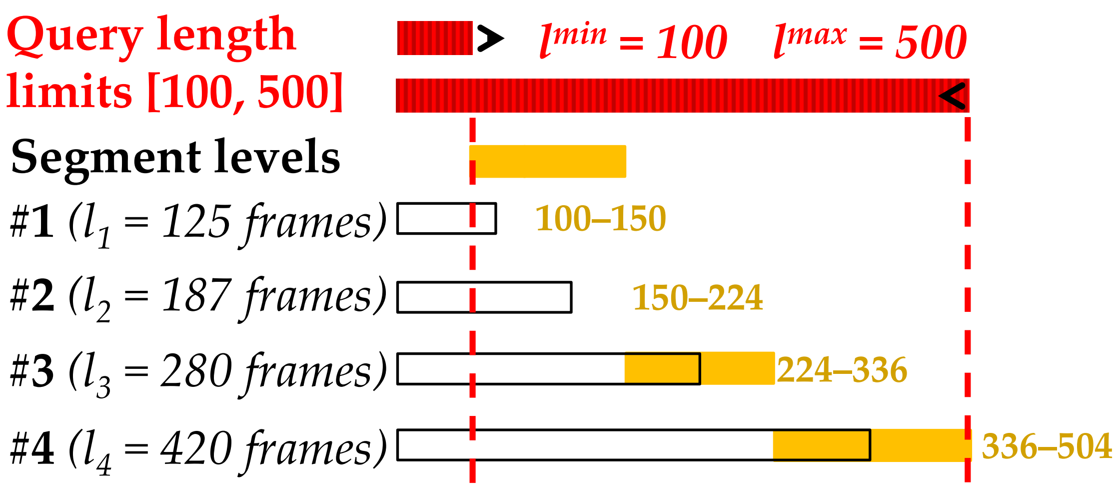


#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits (l^{min}, l^{max})
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity $cf = 20\ \%$:

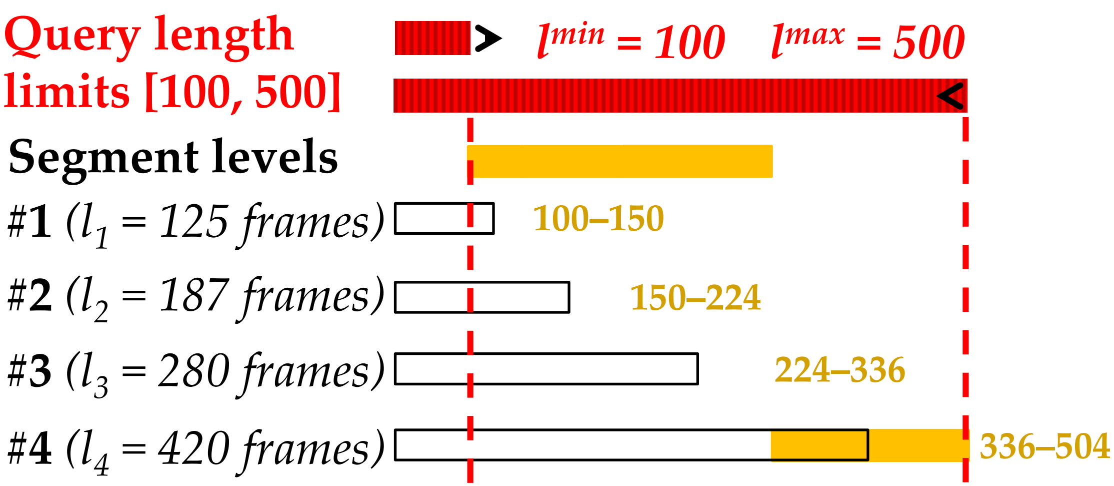


#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:

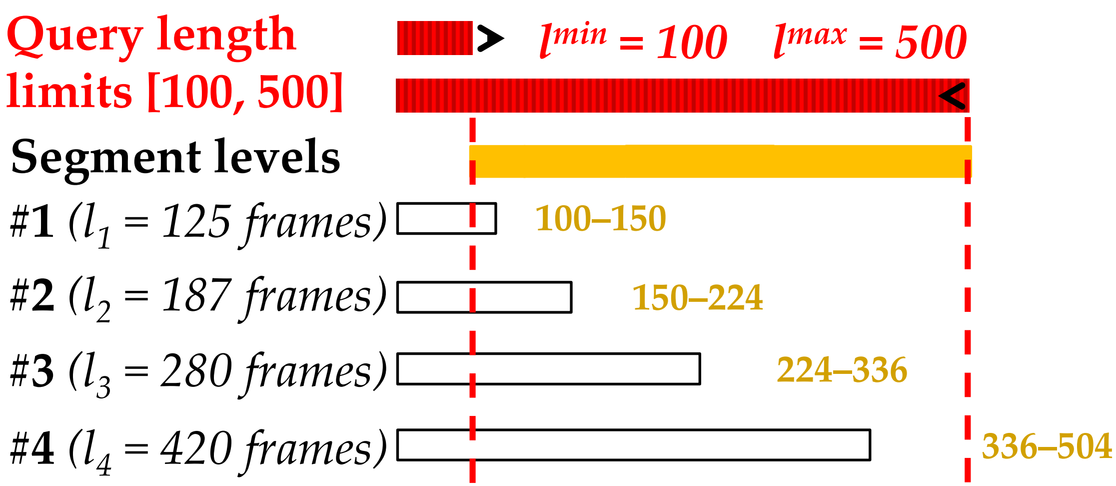


#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:



#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:

Query length limits [100, 500]

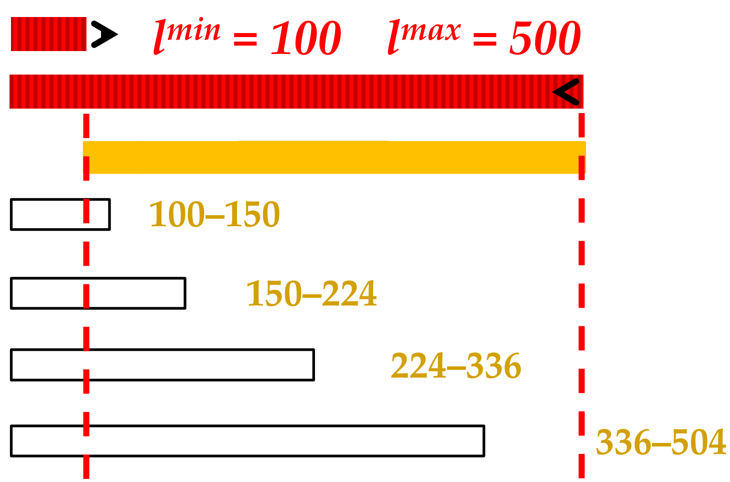
Segment levels

#1 (l_1 = 125 *frames*)

#2 (l_2 = 187 *frames*)

#3 (l_3 = 280 *frames*)

#4 (l_4 = 420 *frames*)



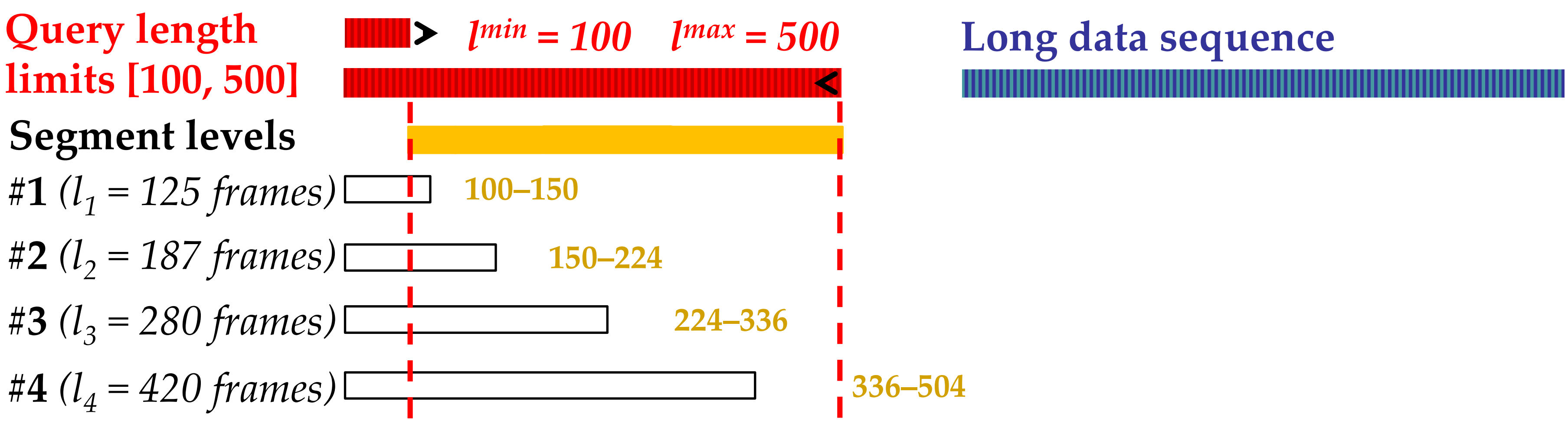
Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:



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

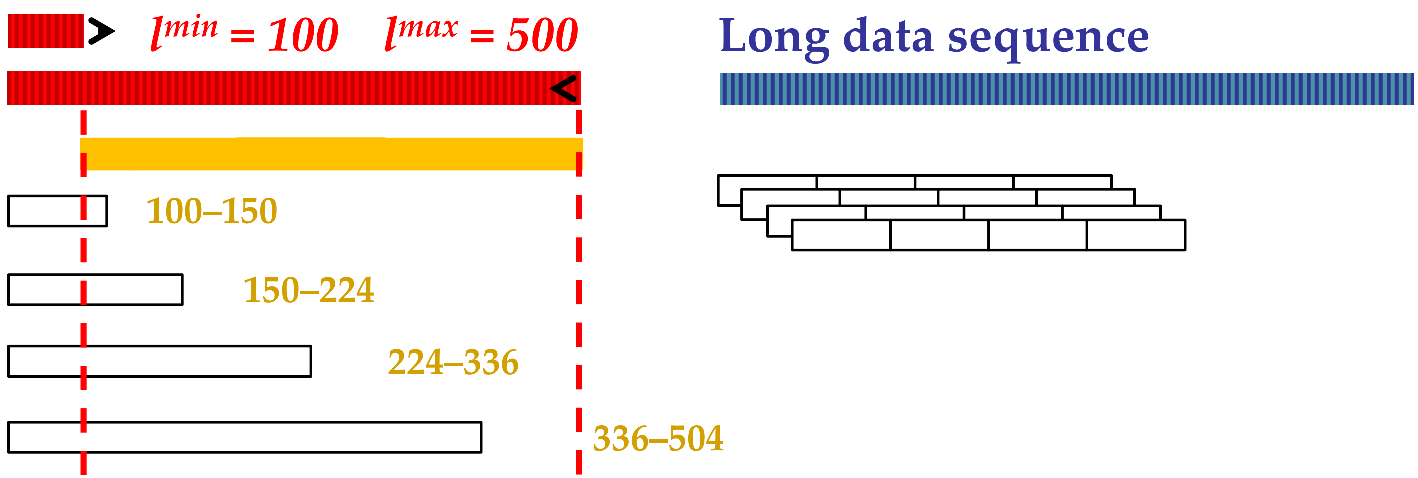
#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 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*)

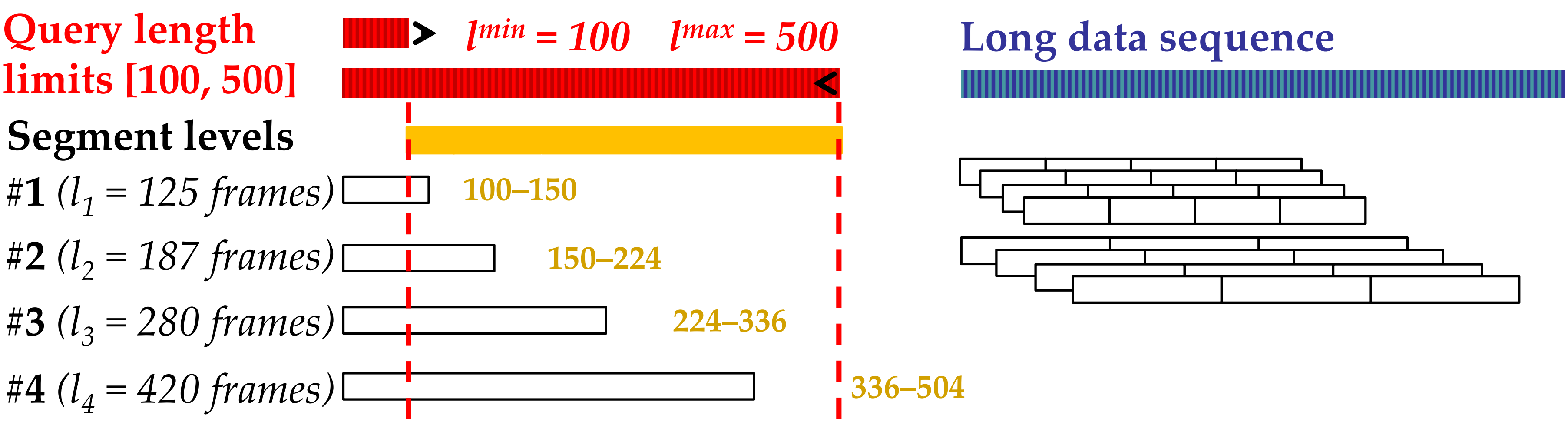
Level shift: $l_n = l_{n-1} * (1 + cf)/(1 – cf)$

#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:



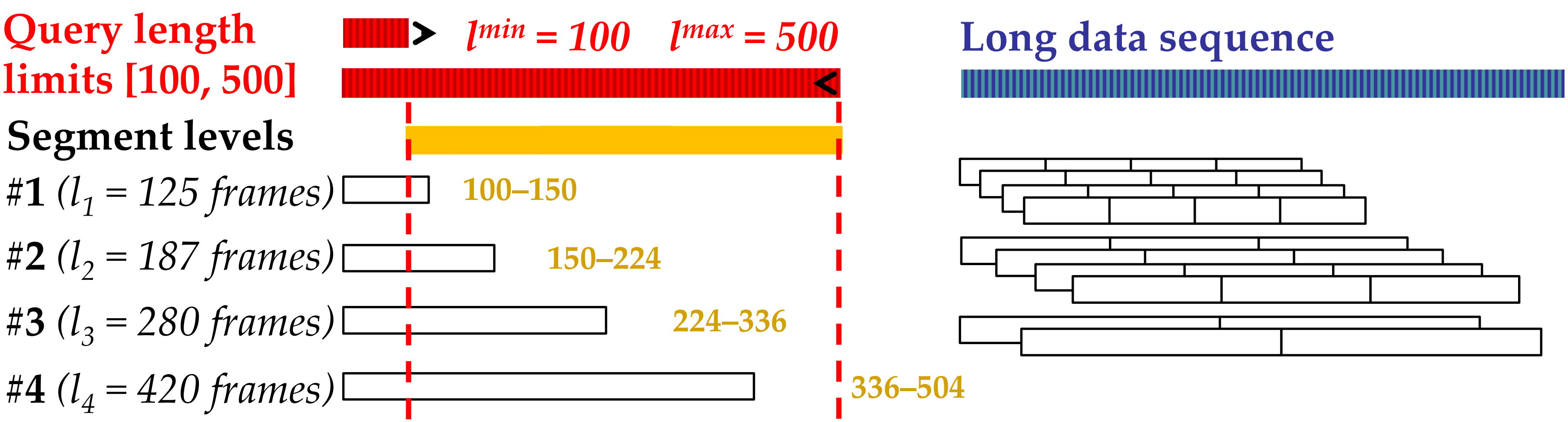
Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:



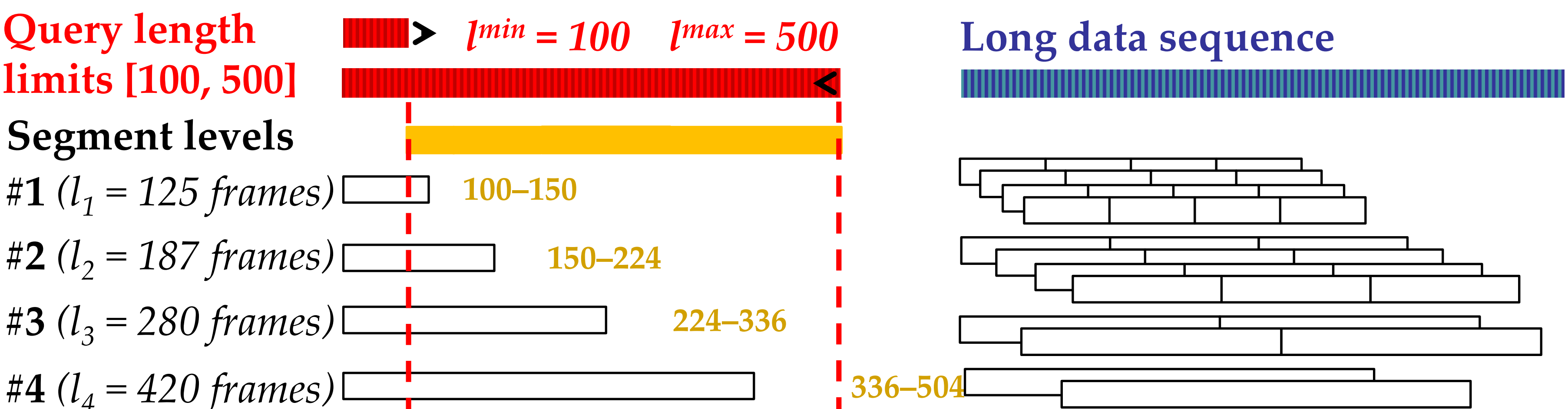
Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:



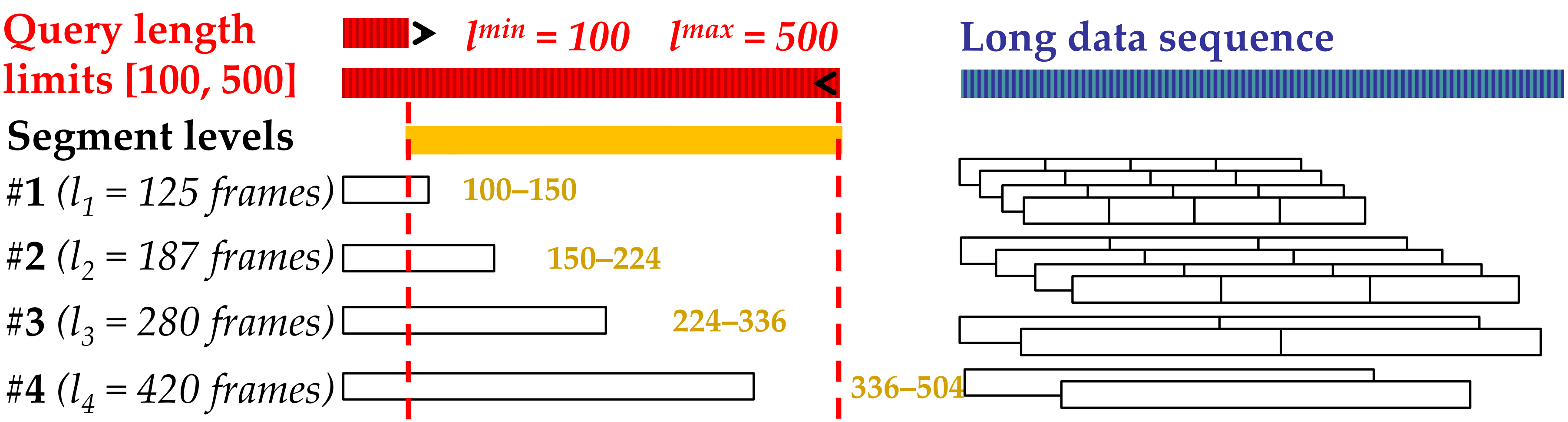
Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

#2 Subsequence Matching – Advanced App.

Advanced multi-level segmentation approach

- Segment lengths and number of levels depend on
  - Query length limits **(l^{min}, l^{max})**
  - Elasticity of the similarity measure (quantified by *cf* parameter)

Segmentation example for elasticity *cf* = 20 %:

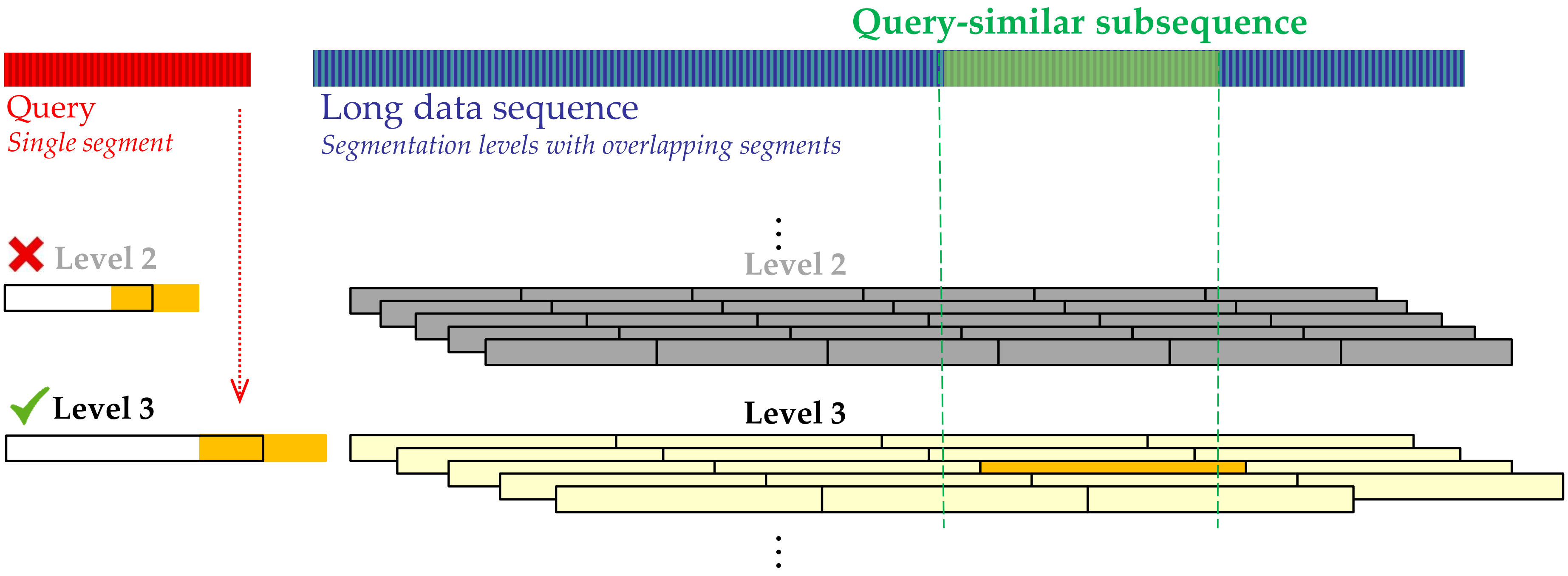


Level shift: $l_n = l_{n-1} * (1 + cf)/(1 - cf)$

Segment shift: $l_n * cf$

#2 Subsequence Matching – Advanced App.

- 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 from at most **100 – *cf*** [%]
- The *k* most similar segments presented as the query result



Segmentation in Numbers

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)

Segmentation in Numbers

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**
SISAP ´16 – naive	160,000,000	120,000	**400,000**
SISAP ’16 – advanced	7,720	20	**1,430**

Experimental Evaluation – Advanced Approach

- **HDM05 Dataset:** 68-minute long data sequence
  - 120 Hz sampling, 31 body joints
  - Ground truth: 1,464 short subsequences in 15 categories (~queries)
- Subsequence retrieval using ***k*-NN** queries:
  - ***l^{min} = 41 frames (340ms), l^{max} = 2,063 frames (17.2s)***
  - **Different settings of elasticity *cf = {10%, 20%, 30%, 40%, 50%}***

cf [%]	# of levels	# of levels		Feature extract. time [min]	Sequential scan [ms]	Precision	
		total	1st level			$k = 1$	$k = 5$
10	18	631,746	111,774	263.2	447	87.30	84.37
20	9	150,971	51,230	62.9	205	86.75	84.13
30	6	66,972	31,526	27.9	126	86.89	82.98
40	5	37,345	21,955	15.6	88	85.79	82.65
50	4	23,669	16,393	9.9	66	84.43	81.99

Conclusions

Advanced subsequence matching in mocap data

- Query always considered as a single segment
- The elasticity property of the similarity measure enables to dramatically reduce the number of data segments

Efficiency

- Searching the 68-minute sequence sequentially takes 205ms
- By applying the PPP-Codes [Novak et al., TLDKS 2016] to index data segments at each level, search times can be further decreased by two orders of magnitude
  - Approximate search within a 121-day long data sequence in 1 second

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