Motion Words: Efficient and Effective Representation of Motion Capture Data

Petra Budíková, Vlastislav Dohnal, Jan Sedmidubský, Pavel Zezula

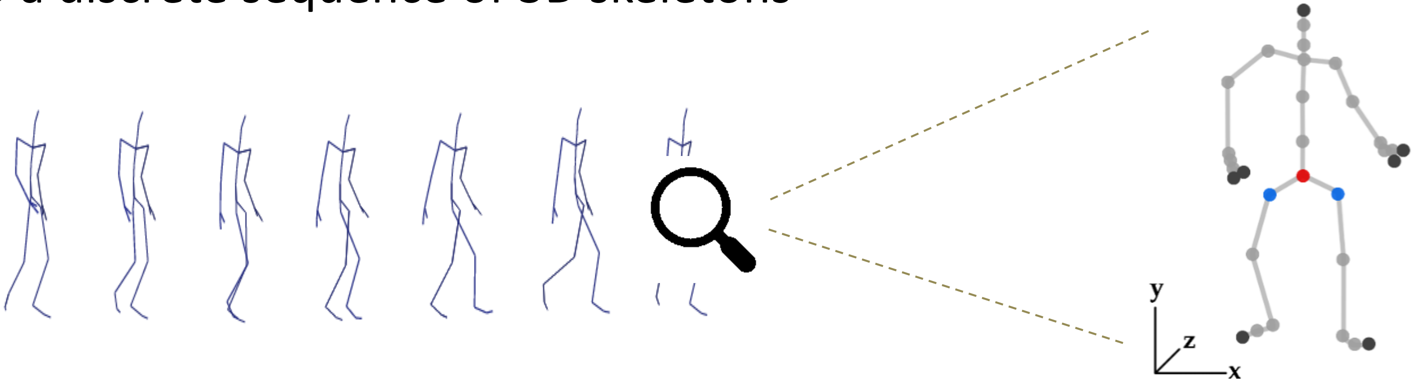
Outline

- WHY motion words?
  - Challenges of motion data processing
  - Limitations of existing approaches
  - Inspiration from related fields
- HOW can motions be represented by motion words?
  - Overview of our approach
  - Discussion of individual steps
  - Preliminary results

WHY motion words?

Motion capture (MoCap) data

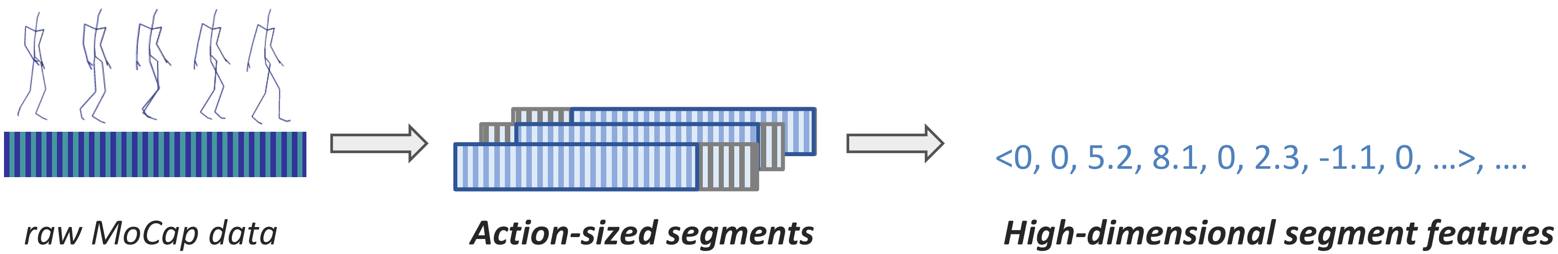
- Continuous spatio-temporal characteristics of a human motion simplified into a discrete sequence of 3D skeletons



- Many application domains: computer animation, medicine, sports, …
- Standard motion analysis operations: classification, subsequence search, semantic annotation
  - Common task: determining similarity of two motion sequences

Evaluating motion similarity

- State-of-the-art: features trained for whole actions

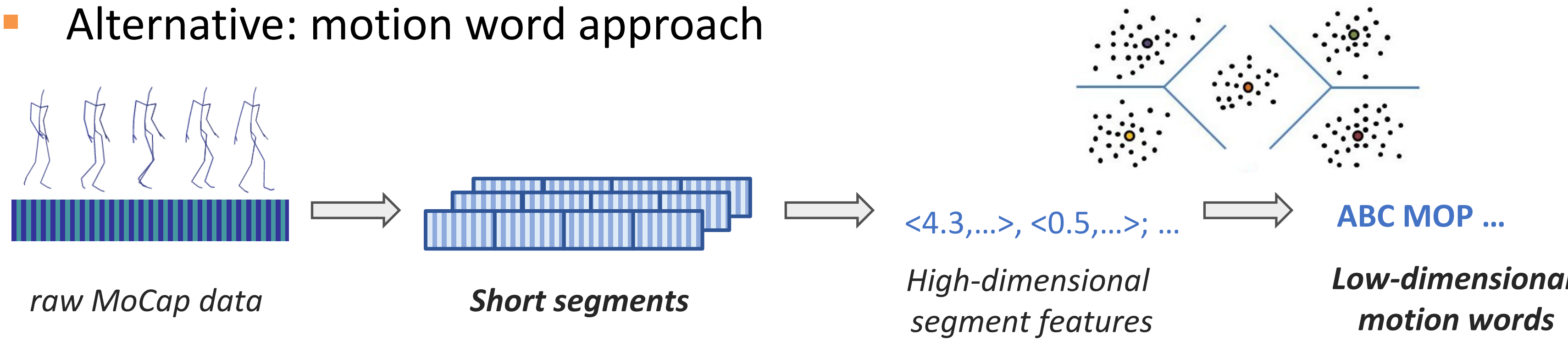


similarity of two motion sequences = similarity of the respective two features

- Advantages:
  - High-precision neural networks can be trained
  - Suitable for action recognition
- Disadvantages:
  - Limited applicability e.g. for subsequence search
    - Typically works for a limited range of segment sizes
    - High memory requirements (data replication) and retrieval costs

Evaluating motion similarity (cont.)

- Alternative: motion word approach

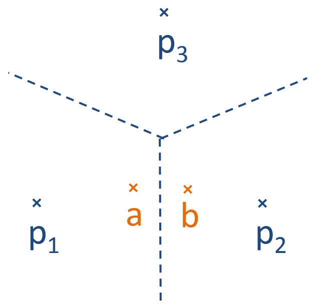


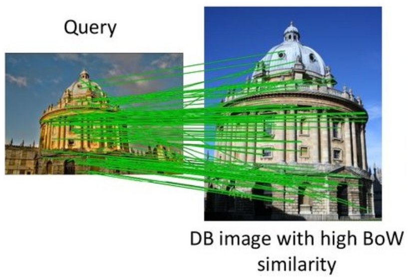
similarity of two motion sequences = similarity of the sequences of motion words

- Expected advantages:
  - Applicable to a wide range of MoCap processing tasks
  - Applicable for comparing motion sequences of any size
  - Compact motion representation, lower memory requirements
  - Efficient text-processing methods can be applied for indexing and retrieval

Inspiration: visual words

- Around 2000, local image descriptors were very popular for image retrieval
  - Effective, but not efficient: a high number (500-3000) of high-dimensional (128 for SIFT) features per single image!
- Josef Sivic, Andrew Zisserman: Video Google: A Text Retrieval Approach to Object Matching in Videos. ICCV 2003.
  - Use clustering to quantize feature descriptors into visual words
  - Apply text-processing techniques
- Many following works:
  - Feature quantization:
    - Trying to overcome efficiency problems:
      - hierarchical k-means, approximate k-means, randomized methods
    - Trying to minimize "border problems":
      - Fuzzy clustering (weighted combination of several visual words for each feature)
      - Consensus clustering (multiple visual vocabularies, different levels of consensus)
  - Spatial verification of candidates





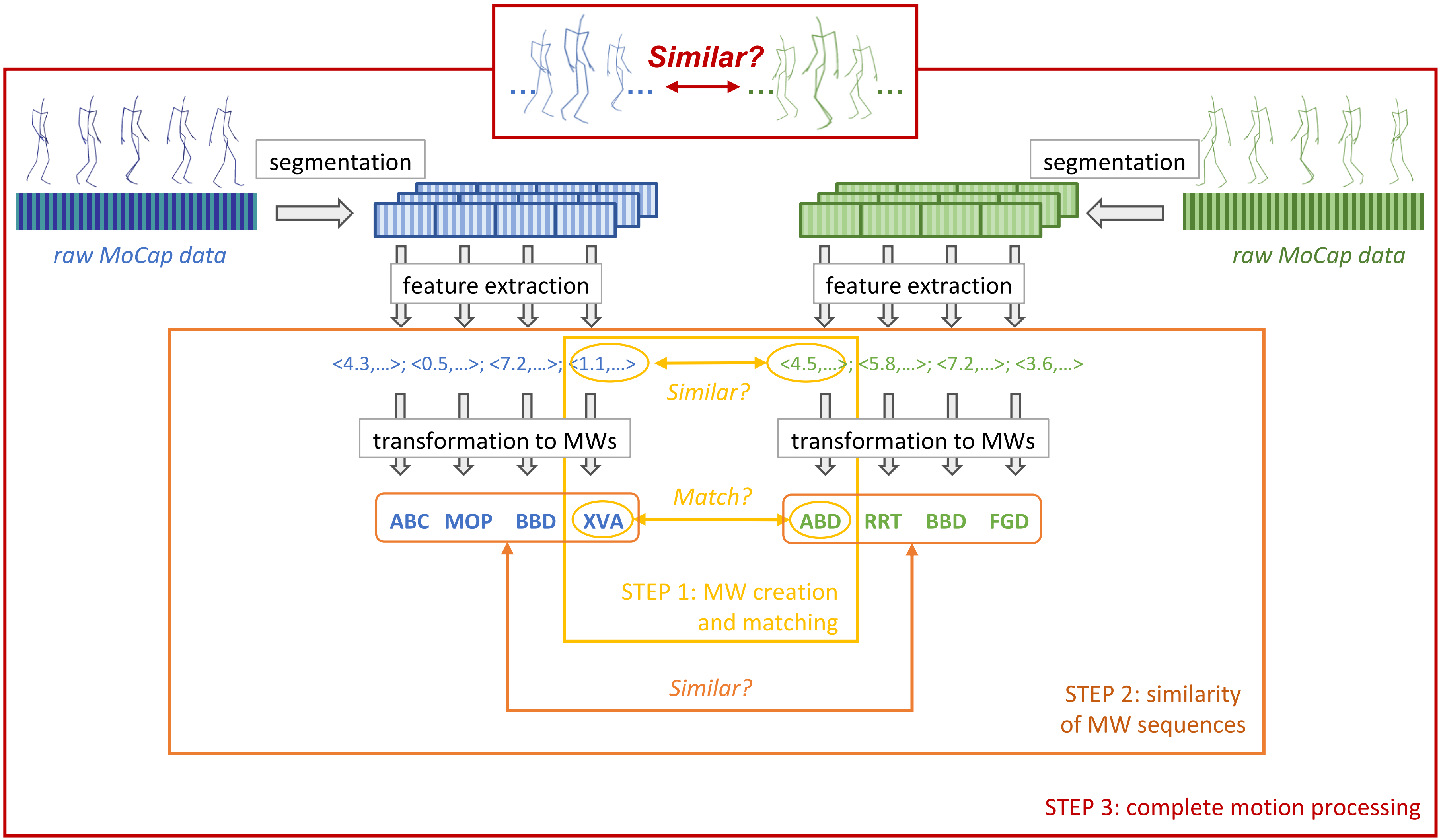
Similar ideas in motion processing

- Rongyi Lan, Huaijiang Sun: Automated human motion segmentation via motion regularities. The Visual Computer 31(1): 35-53 (2015)
  - Cluster individual poses into motion words
    - Agglomerative hierarchical clustering
  - Apply probabilistic modeling to discover motion topics

- Aristidou, A., Cohen-Or, D., Hodgins, J. K., Chrysanthou, Y., & Shamir, A. (2018). Deep Motifs and Motion Signatures. In *SIGGRAPH Asia 2018*
  - Break motion sequences to short-term movements called *motion words*
  - Cluster the motion words into *motion motifs*
    - K-means clustering algorithm, mutually exclusive clusters
  - The *signature* of a motion sequence S is defined as the normalized histogram of its words in all K clusters.
    - For comparisons, use tf-idf weighting and Earth Mover's Distance

Motion words – HOW?

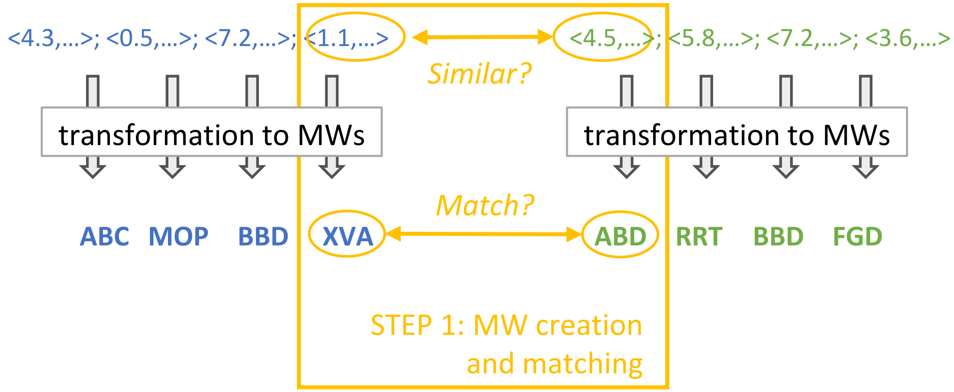
Processing with MWs: overview



Our objectives

- Demonstrate the viability of the MW approach
  - Propose solutions for all phases
  - Show that together they work in a real-world scenario
    - With reasonable quality
    - With high efficiency and scalability (at least in theory)
- Identify problems, provide insight into individual steps using real data
  - There are multiple phases where we can lose information
    - Segmentation, feature extraction, quantization, matching
  - We want to understand the influence of individual techniques, therefore we would like to evaluate each step independently

Step 1: MW creation and matching

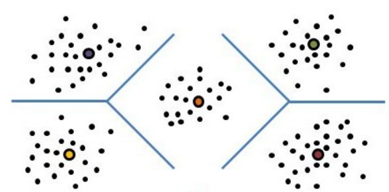


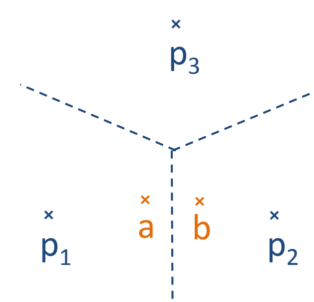
- Input: segment features and distance function
- Output: motion words and MW matching function

- What do we want?
  - segments similar in the original feature space will be matched in the MW representation
  - dissimilar segments will not be matched

Towards formalization of MWs

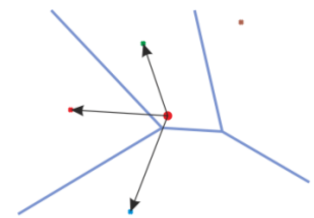
- Motion word (basic version)
  - One-dimensional representation of MoCap data segment
  - Obtained by disjoint quantization of the original MoCap data (features and distance measure)
    - Each motion segment is associated with one MW
  - Coarse approximation of the original MoCap similarity function by trivial MW matching function:
    - segments that are mapped on the same MW have similarity 1
    - segments that are mapped different MWs have similarity 0
- Motion word vocabulary
  - Set of available MWs defined by a particular quantization technique
  - Can be seen as a set of equivalence classes over the original feature space
- Problems:
  - Assumes one optimal c
  - Border problems are very likely to occur





Towards formalization of MWs (cont.)

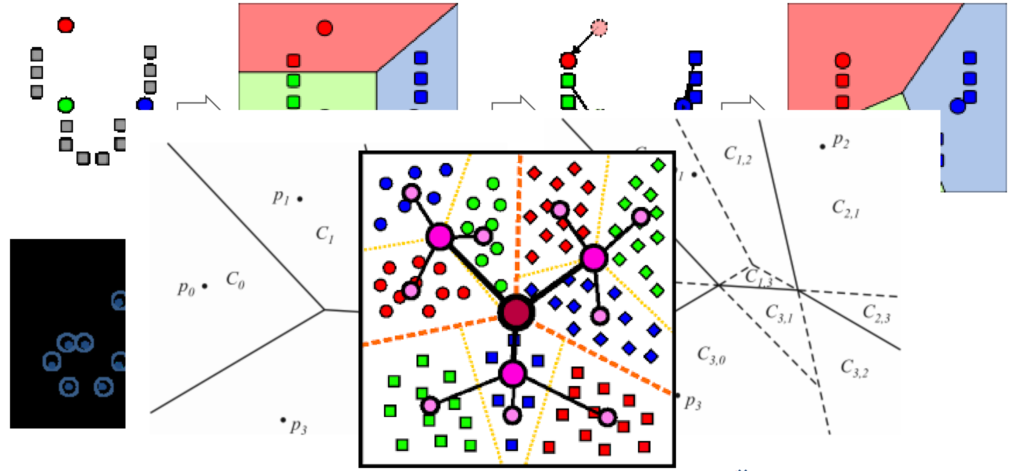
- Motion word (generalized version)
  - One-dimensional representation of MoCap data segment
  - Obtained by soft (fuzzy, overlapping) quantization of the original MoCap data (features and distance measure)
    - Each motion segment is associated with one or several motion words, potentially with confidences
      - Segment s1 -> motion words {A,B,C}
      - Segment s2 -> motion words {B,C,X}
      - Segment s3 -> motion words {C,X,Y}

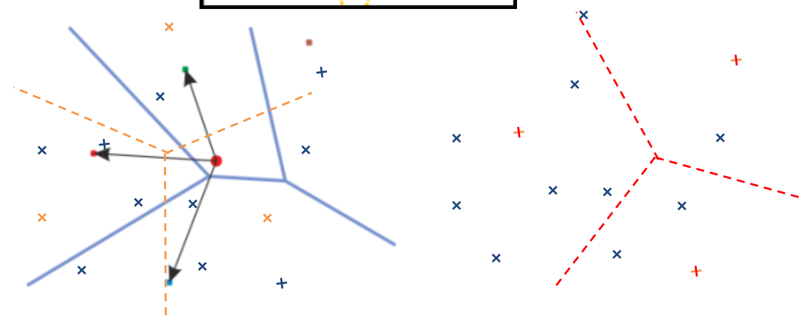


  - Non-trivial MW matching function
    - Motion segments are considered similar if all/some/at least *k* of their MWs match
      - Not transitive, does not define equivalence classes
      - Should provide better approximation of the original similarity between motion segments
- Motion word vocabulary
  - Set of available MWs defined by a particular quantization technique
  - Motion words may not be equivalence classes over the original feature space
    - Motion word A: {s1}
    - Motion word B: {s1,s2}
    - Motion word C: {s1,s2,s3}

Quantizing features into MWs

- Hard clustering
  - Flat partitional clustering
    - *k*-means clustering
  - Hierarchical clustering
    - Divisive
      - Hierarchical *k*-means
      - M-index
    - Agglomerative
- Soft clustering
  - Fuzzy assignment to clusters
    - *k* nearest clusters
    - All clusters with close borders
  - Consensus clustering





- Things to consider:
  - Vocabulary size = number of clusters
    - Text retrieval: hundreds of thousands for full language dictionary
    - Visual retrieval: hundreds of thousands or millions
    - Motion retrieval: ???
      - In *Deep Motifs and Motion Signatures* they use 100 motifs

MW matching

- **Trivial MW matching function:** $MW \times MW \rightarrow \{0,1\}$
  - only equal MWs match

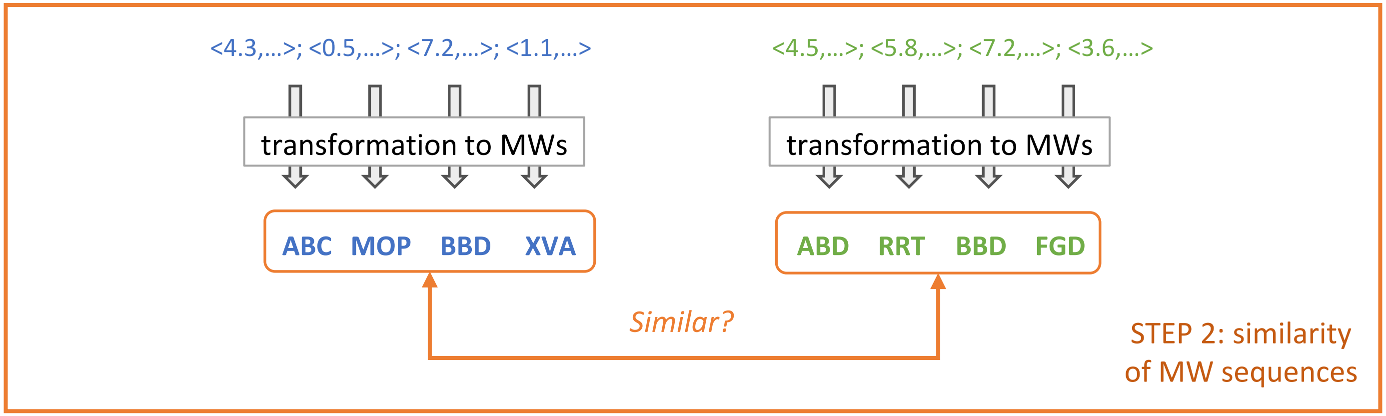
- **Non-trivial MW matching function:**
  - If we do not assume MW confidences: $2^{(MW)} \times 2^{(MW)} \rightarrow \{0,1\}$
    - Two sets of MWs match if the cardinality of their intersection is at least *n*
  - With MW confidences (fuzzy clustering):
    $$2^{(MW \times confidence)} \times 2^{(MW \times confidence)} \rightarrow \{0,1\}$$
    - Future work

Evaluation of MW matching

- Standard cluster evaluation
  - External – compares given clustering C to GT clustering C_{GT}
    - Rand index: probability that C and C_{GT} will agree on a random pair of objects
  - Internal – no GT, uses intra- and inter-cluster distances
    - Silhouette coefficient: measure of how similar an object is to its own cluster (cohesion) compared to the neighbor cluster (separation)

- Unfortunately, there is no external GT for segment matching
  - However, we can use the distribution of distances in the original feature space to define a partial approximate GT clustering $C_{GT\text{-}approx}$
    - If $dist(o_1,o_2) <= dist_{SIMILAR}$, then o_1 and o_2 belong to the same cluster in $C_{GT\text{-}approx}$
    - If $dist(o_1,o_2) > dist_{DISSIMILAR}$, then o_1 and o_2 belong to different clusters in $C_{GT\text{-}approx}$
  - Using $C_{GT\text{-}approx}$, we can define "semi-external" evaluation measures
    - E.g. Unsupervised Rand index

Step 2: similarity of MW sequences



- Input: MW sequence and MW matching function
- Output: MW sequence distance function

- What do we want?
  - Depends on application
    - Find very similar motions different only in speed
    - Find similar motions with gaps
    - Detect longer sequences with similar subsequences
    - …
  - Common requirement: reasonable distribution of distances in the dataset

Sequence similarity

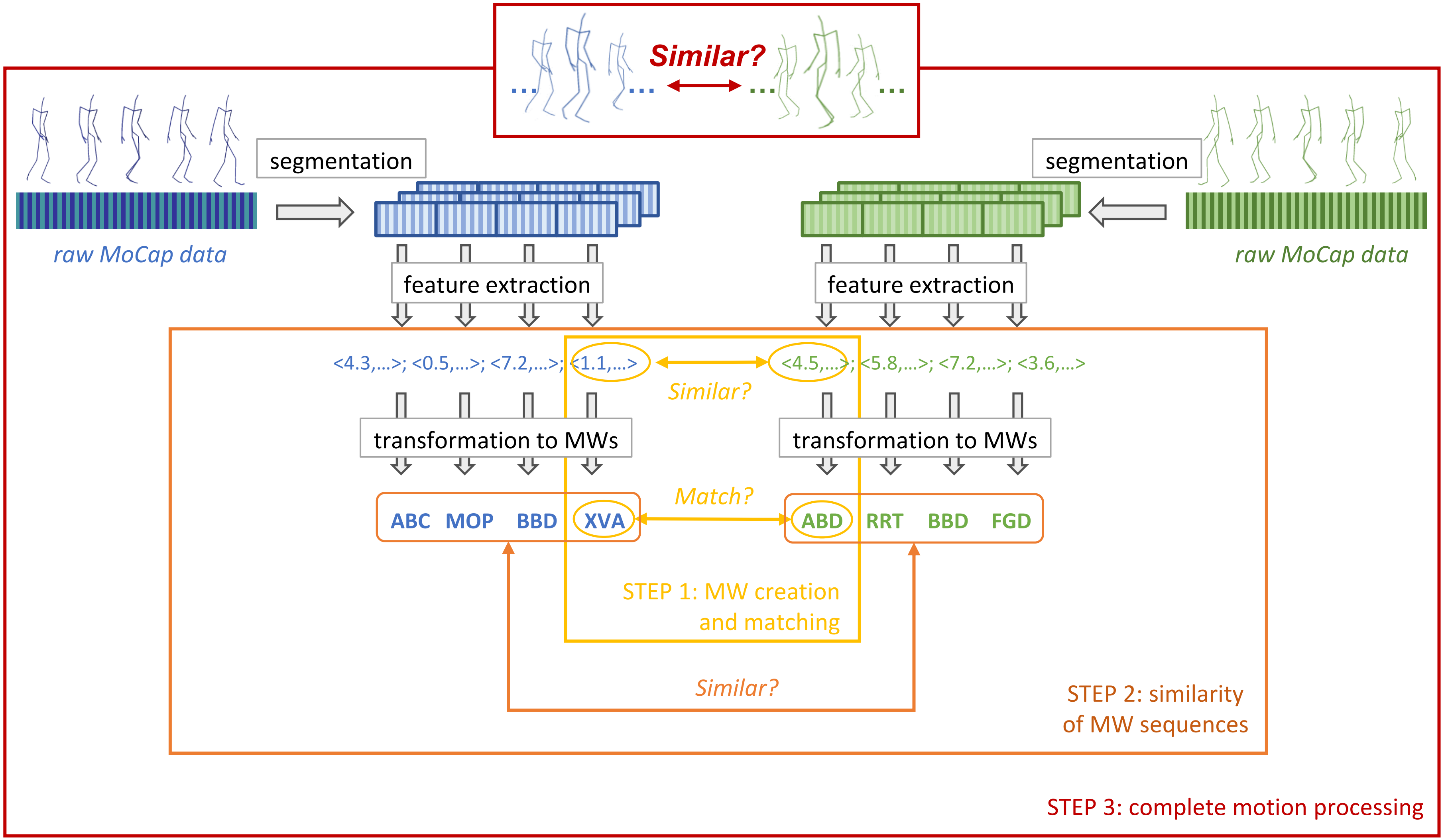
- Possible approaches:
  - Set of words
    - Jaccard similarity
  - Bag of words (histograms, vectors)
    - Euclidean distance
    - Cosine distance
    - Earth movers distance
  - Sequence matching
    - Edit distance
    - DTW
    - Sequence alignment
    - Longest common subsequence
    - Shingles + Jaccard similarity

Sequence similarity (cont.)

- Things to consider:
  - Word weighting
  - Stop words
  - Efficient indexing!

- Evaluation
  - Look at distance distribution of MW sequences

Step 3: complete motion processing with MWs



Complete motion processing with MWs

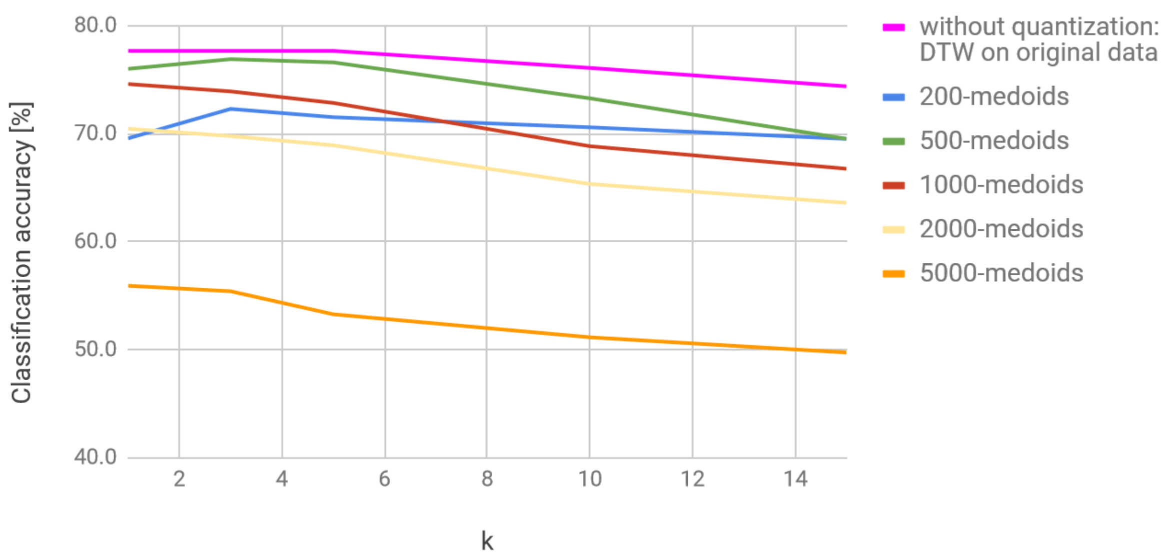
- With respect to a given application, choose suitable segmentation, features, quantization, matching, sequence similarity

- Segmentation
  - Static or semantic?
    - Now: static
    - Future work: try semantic segmentation
  - What is reasonable segment length?
  - Disjoint or overlapping segments?

- Segment features
  - Now: original 3D data + DTW
  - Future work: better segment features
    - Train NN?

Preliminary results

- Application: action recognition
  - 130 classes, 2345 actions
  - kNN classifier
- Settings:
  - Static segmentation, segment length 80 frames, shift 16 frames
  - Segment features: original 3D data + DTW
  - Feature quantization: flat k-medoids
  - Similarity evaluation: trivial MW matching, DTW for MW sequence similarity



The final slide (recap)

- To make the MW idea work, we need to solve:
  - Step 1: MW creation and matching
  - Step 2: similarity of MW sequences
  - Step 3: complete motion processing with MWs

- What we have:
  - First simple solution that provides not-so-bad results
  - A lot of avenues to explore:
    - Soft clustering methods
    - MW sequence similarity measures
    - Different segmentation strategies