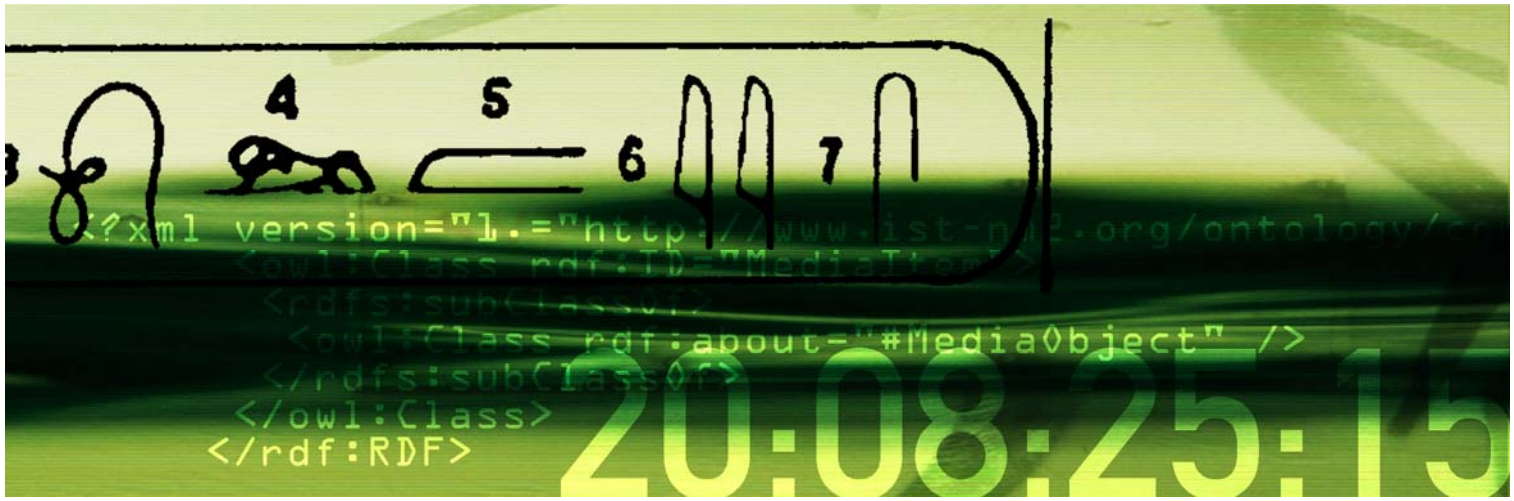
JOANNEUM RESEARCH

K-Space

www.joanneum.at

Institute of Information Systems & Information Management

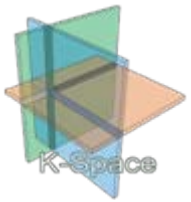


Eliminating the Back-Tracking Step in the Longest Common Subsequence (LCSS) Algorithm for Video Sequence Matching

Werner Bailer

SAMT 2008, Koblenz, 2008-12-04

JOANNEUM RESEARCH



Outline

- Motivation
  - detecting repeated video content
- Core Problem: distance between video sequences
- Proposed distance measure
- Optimisation
- Demo

Core Problem: Measuring Distance between Video Sequences

- Requirements
  - support any type of features (audio/video, discrete or continuous values, scalar/vector, custom distance functions between two feature values/vectors)
  - support partial match of sequences
  - different timing of sequences
  - support gaps and insertions in the matching parts
  - minimum length of useful match
- Approaches
  - Dynamic Time Warping: assign each element of sequence A to nearest element of sequence B
  - String edit distance: identical, replace, insert, delete

Distance Measure for Video Sequences (1)

- match sequences of feature vectors extracted from the input videos
- based on Longest Common Subsequence (LCSS) algorithm
  - variant of string edit distance
  - build matrix from matches
  - find longest matching subsequence which may have gaps by back-tracking
  - original LCSS algorithm assumes discrete input values, match := equality of input values
- LCSS for vectors of continuous values
  - proposed for 2D trajectories [Vlachos et al., 2002]
  - match := Euclidian distance between elements < ε and offset between elements < δ

Distance Measure for Video Sequences (2)

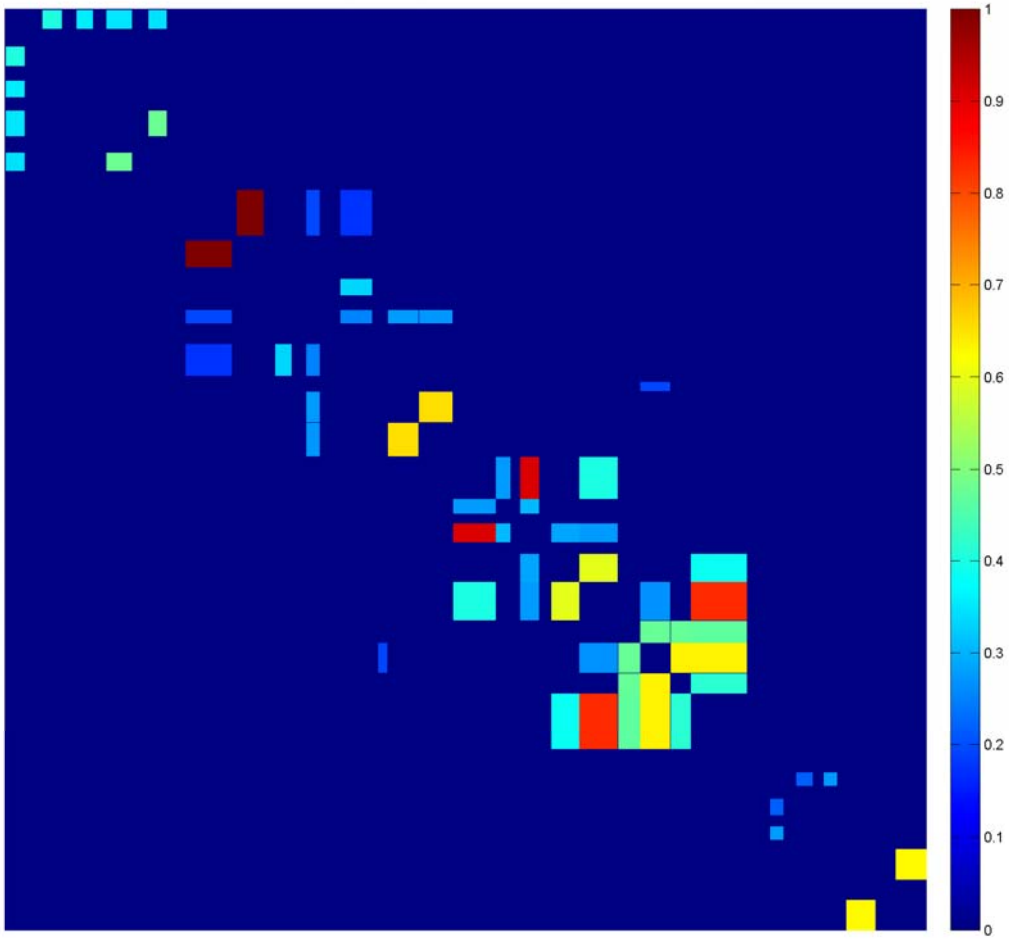
- for matching feature sequences of videos
  - replace ε by a vector of thresholds $\theta_{sim}=\{\varepsilon_1,...,\varepsilon_F\}$ for m features 1,...,*F*, which are weighted by weights $\{w_1,...,w_F\}$
  - match features *f* with appropriate distance functions
  - discard δ, absolute temporal distance of feature vectors in the sequence is irrelevant
  - accept all matches longer than a minimum length θ_{len}
  - introduce maximum gap γ between two consecutive matching feature vectors
  - consequence of gap constraint:
    - not just *longest* common subsequence (might have gaps)
    - but all *sufficiently long* ($> \theta_{len}$) subsequences
  - similarity := length of match / min(lenghts of input sequences)

Distance Measure for Video Sequences (3)

- 1D Example
  - input sequences top/left
  - $\theta_{sim} = 0.5$
  - $\theta_{len} = 3$
  - $\gamma = 1$

	2.3	4.7	1.2	3.3	2.2	1.4
1.3	0	0	1	1	1	1
2.5	1	1	1	1	2	2
3.4	1	1	1	2	2	2
2.4	1	1	1	2	3	3
4.6	1	2	2	2	3	3
1.5	1	2	3	3	3	4
2.6	1	2	3	3	4	4

Distance Measure for Video Sequences (4)

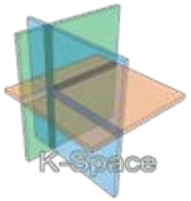


Distance Measure – Optimisation: Eliminating Back-Tracking

- proposed optimisation
  - keep while building matrix
    - last match in each line and list of current sequences
  - for each match
    - find nearest previous match (city block distance – 2)
    - search in $-(\gamma + 1)$ lines, add to this sequence: $O(1)$

	2.3	4.7	1.2	3.3	2.2	1.4
1.3	0	0	1	1	1	1
2.5	1	1	1	1	2	2
3.4	1	1	1	2	2	2
2.4	1	1	1	2	3	3
4.6	1	2	2	2	3	3
1.5	1	2	3	3	3	4
2.6	1	2	3	3	4	4

JOANNEUM RESEARCH



Demo

- Video Browsing Tool
- Rushes Skimming