Recognition of Manual Actions Using Vector Quantization and Dynamic Time Warping

Marcel Martin Jonathan Maycock
Florian Paul Schmidt Oliver Kramer

technische universität
dortmund

CITEC
Cognitive Interaction Technology
Center of Excellence
Bielefeld University

ICSI
INTERNATIONAL
COMPUTER SCIENCE
INSTITUTE

Overview

Task

Given recording of hand movements, automatically recognize where known manual actions start and end.

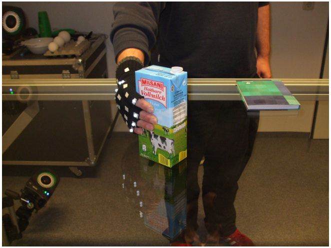
We . . .

- use modified Dynamic Time Warping to locate and classify actions in one step
- use vector quantization to reduce computation time

Pipeline

1. Motion capture
2. Feature computation
3. Modified Dynamic Time Warping
4. Vector quantization

Motion capture setup



- 14 Vicon cameras track 11 reflective markers on one hand
- 20 samples/second
- Vicon software outputs coordinates of each marker

Input for our our algorithm: 33-dimensional time series data

Recordings

Reference manual actions

- Approx. 2.5–5 seconds each
- Move cup to mouth, write in book, pick up cup, . . .

Action sequences

- Concatenation of $\approx$ 10 manual actions ($\approx$ 30 seconds)
- First pick up pen, then write in book, then put down pen, then . . .

Euclidean coordinates cannot be used directly

Requirement

Same motion made at different locations or towards different directions must be recognized as same

Some suitable features

- Inclination of the hand: angle between the table and the axis through the index finger knuckle and baby finger knuckle
- Distance between the thumb tip and the index finger tip
- Velocity of the z-coordinate of the barycenter (center of mass)
- Average distance of five finger tips to barycenter

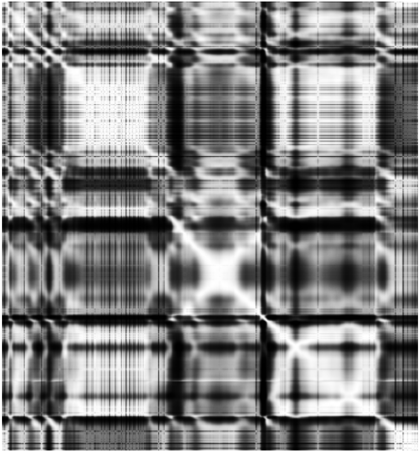
Modifying Dynamic Time Warping to locate actions

First:

Regular Dynamic Time Warping (DTW)

- Compares two recorded signals from start to end, allowing distortions on time axis
- Outputs a distance

DTW example



Shown here

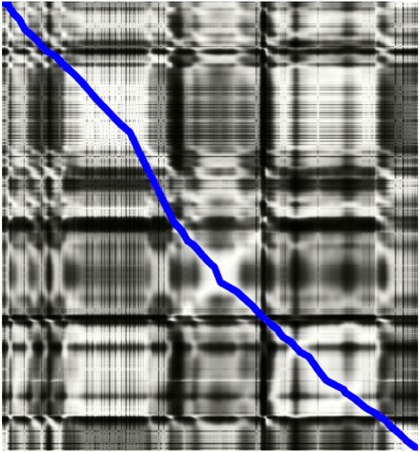
Comparison of all vectors of one sequence to all of other sequence.

White pixels: Low euclidean distance (vector norm)

DTW example

action sequence 1

action sequence 2



Shown here

Comparison of all vectors of one sequence to all of other sequence.

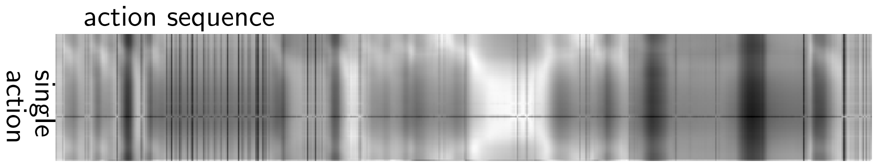
White pixels: Low euclidean distance (vector norm)

Blue: Path with lowest sum of distances (DTW distance)

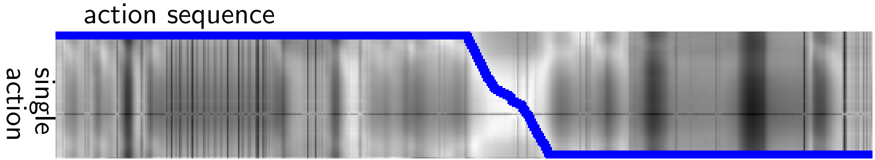
Modification to Dynamic Time Warping

- Inspiration from bioinformatics: They use “semi-global” alignment to find a pattern within text
- Idea: Add (conceptual) “white” row above and below matrix
- $\rightarrow$ sequence 2 may now start and end anywhere within sequence 1
- Output: DTW distance **and** location of searched sequence

Example of modified DTW

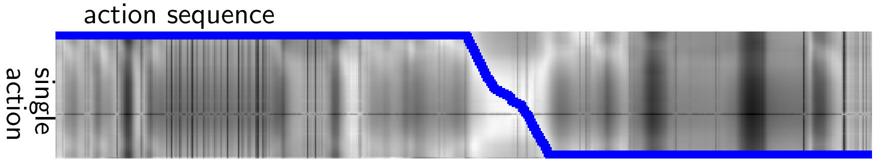


Example of modified DTW



Blue: Path with lowest sum of distances

Example of modified DTW



Blue: Path with lowest sum of distances

Remaining problem

All $n \cdot m$ euclidean distances must be computed

(n: length of action sequence, m: length of reference action)

Speed up computation using symbols

Input data: Sequences of feature vectors from $\mathbb{R}^d$

- Idea: Map each feature vector to a **symbol from a finite set**
- Similar vectors are mapped to the same symbol
- Only work with sequences of symbols (string)

Advantage: Distance computation is **faster**

- Compute distance between two d-dimensional vectors: $\mathcal{O}(d)$
- Compute distance between two symbols (table lookup): $\mathcal{O}(1)$

Vector quantization (VQ)

Our definition

Mapping a large set of vectors to a small set of vectors that are representative for the large set

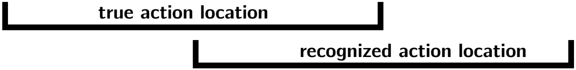
Use an unsupervised learning method, such as k-means, Growing Neural Gas (GNG), self-organizing map (SOM):

1. Train classifier on set of all feature vectors
2. Classify each vector into one of k classes
3. Replace each feature vector with its class label $(1, \ldots, k)$
4. Additionally: Pre-compute distances between all classes

Evaluation

Algorithm:

1. Compute features from input data
2. Train vector quantization algorithm
3. Convert sequences to symbols
4. For each action sequence and for each reference action: Run modified DTW and determine **overlap** of true and predicted intervals.



5. Output average overlap

Results

Test data

14 action sequences containing 132 known actions

Results

	avg. overlap	total runtime [s]	DTW time only [s]
No VQ	**70.4 %**	148.2	133.2
GNG	68.9 %	43.6	**17.5**
SOM	72.2 %	231.0	**16.3**
k-means	70.45 %	41.0	**25.1**

- Almost no decrease of recognition accuracy
- Severe reduction of time spent on DTW

Summary

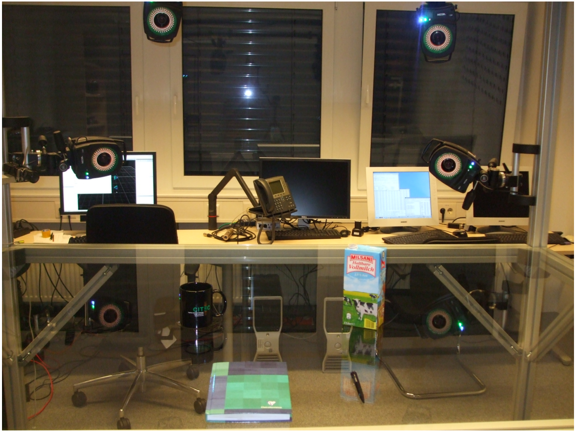
- There are simple features of hand movements that are rotation and translation invariant
- A modification of Dynamic Time Warping can locate and classify manual actions in one step
- Vector quantization can reduce high-dimensional time series data to sequences of symbols
- Using symbols instead of full $\mathbb{R}^d$ vectors reduces DTW computation time
- Using symbols **does not** decrease recognition accuracy

Summary

- There are simple features of hand movements that are rotation and translation invariant
- A modification of Dynamic Time Warping can locate and classify manual actions in one step
- Vector quantization can reduce high-dimensional time series data to sequences of symbols
- Using symbols instead of full $\mathbb{R}^d$ vectors reduces DTW computation time
- Using symbols **does not** decrease recognition accuracy

Thank you for your attention

Motion capture setup



Manual action: Pour milk into cup