Recognizing human motion using eigensequences

Andrea Bottino, Matteo De Simone, Aldo Laurentini

DAUIN - Politecnico di Torino

Motion recognition

- Important topic in CV, many promising applications
  - Entertainment
  - HCI
  - Automatic video indexing
  - Video surveillance
  - ...
- Speed is important (real-time recognition rates are needed)

Contribution of the paper

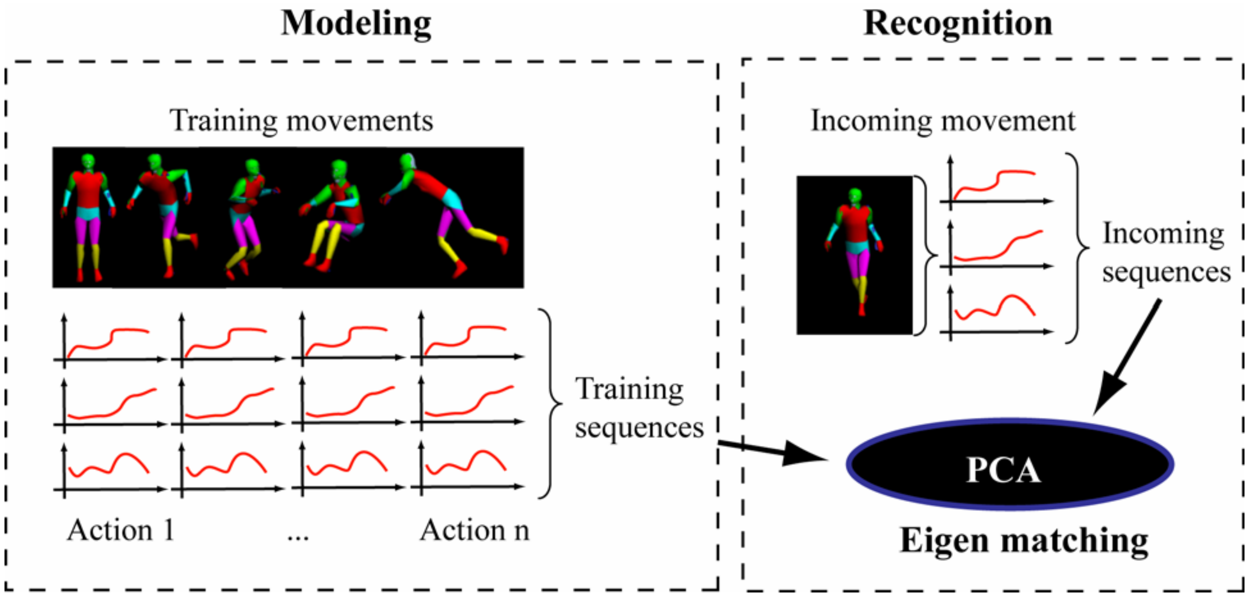
- Real time motion recognition based on 3D model-based motion data
- The basic idea is the following
  - – A *movement* is a curve in the parameter space
  - – These curves are specific of the type of *action* performed
  - – Comparing the whole curves is not a good idea, but we can extract small segments for comparison
  - – In our idea, these segments (*sequences*) are still characteristic of the kind of action performed
  - – PCA techniques allow for an “on-line” recognition

Contribution of the paper

- Extraction of 3D motion data and their classification are actually two independent processes, and this work focuses on motion recognition only.
- Several real-time non-intrusive motion capture techniques are available in literature and can be coupled with our motion classifier
- This paper is aimed at demonstrating the capabilities of the proposed approach. Since, in principle, the recognition process is independent on how motion data have been acquired, we initially tested the approach with motion data acquired with an optical capture system
- We also experimented the proposed approach with a non-intrusive tracker in a virtual environment

Survey of the approach

- Analysis of 3D motion data of a performer and identification of the actions performed
- Motion curves in the parameter space are split into *sequences* (segments of fixed length)
- PCA based representation of sequences is used for recognition



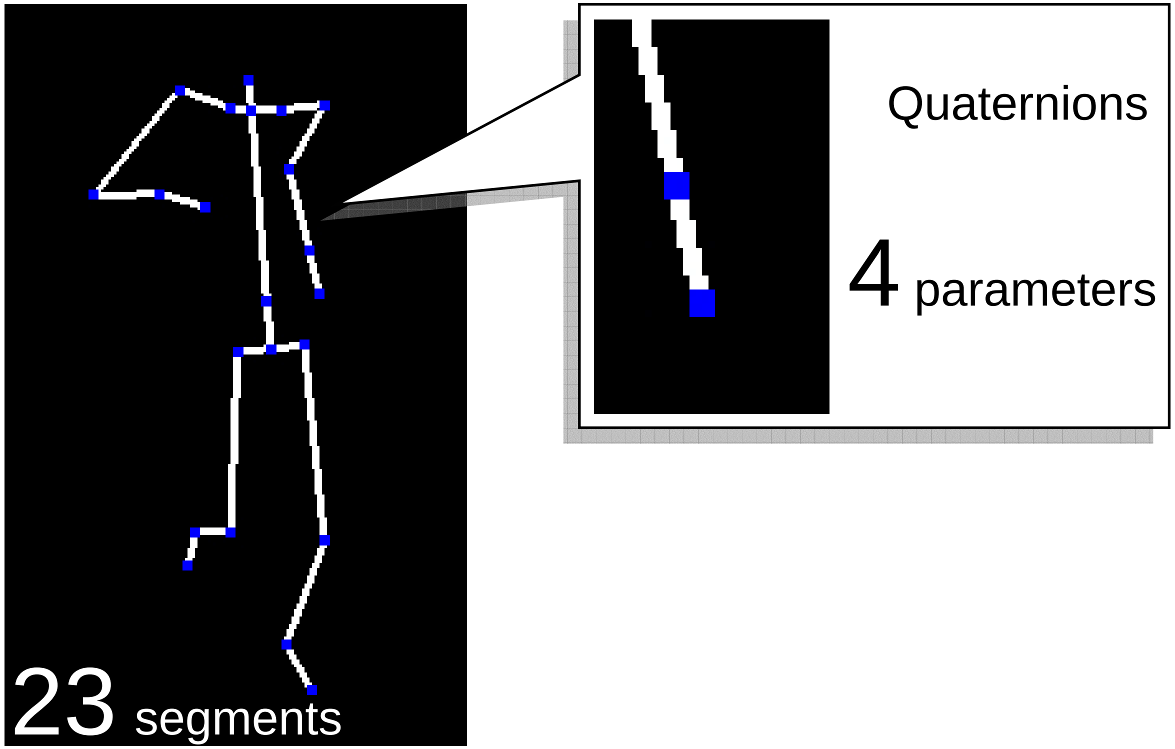
Some definitions

- The attitude of the human body is referred to as a ***posture***
- A ***motion*** is a sequence of contiguous postures over time
- An ***action*** is a specific type of motion, for instance walking, running or sitting

- The aim of our approach is to automatically segment a motion, classifying the various actions it contains

Motion representation

- To represent and reconstruct the motion of a human performer, we use a human body model that is defined by an articulated structure



Quaternions

4 parameters

23 segments

DOF = 26 (rotations) + 3 (pelvis position) = **29**

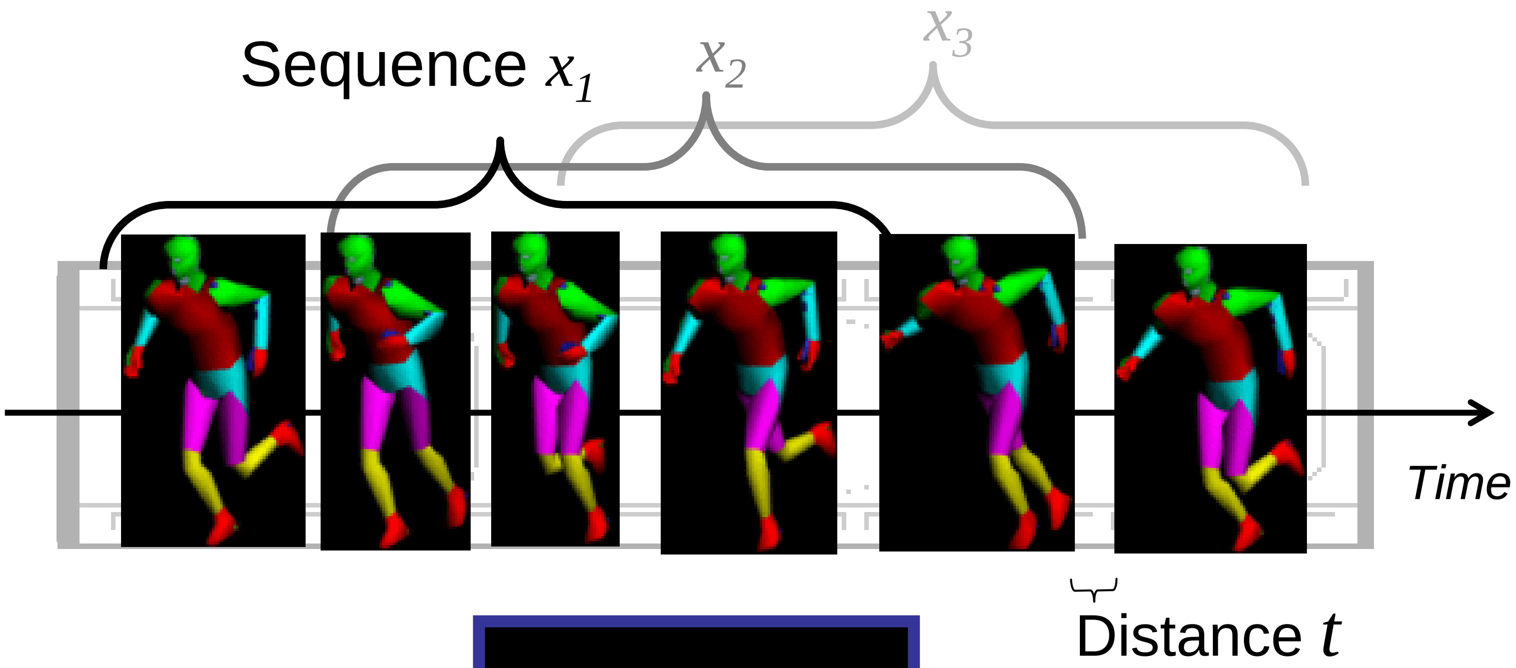
Data normalization

- 3D data from different performers ➔ spatial normalization
- Motion data indipendent from performer’s characteristics and from motion orientation
  - model dimensions are not taken into consideration
  - skeleton’s root translated into the origin of the reference system
  - model is rotated in order to make the pelvis segment coincident with the z axis of the world coordinate systems, heading towards the x axis
  - Dimension of the posture vector reduced to 26
- Temporal normalization?
  - The speed of a gesture is a characteristic of an action
  - Temporal normalization requires the acquisition of the full sequence, preventing real-time recognition

Recognition

- Motion curves for recognition. How?
- Our approach:
  - Splitting a complex movement into a set of “atomic” motions
  - We define as sequence a set of *n* consecutive postures (frames), the starting postures (frames) of two sequences being at a distance of *t* postures (frames)

Sequences



Animation

Recognition

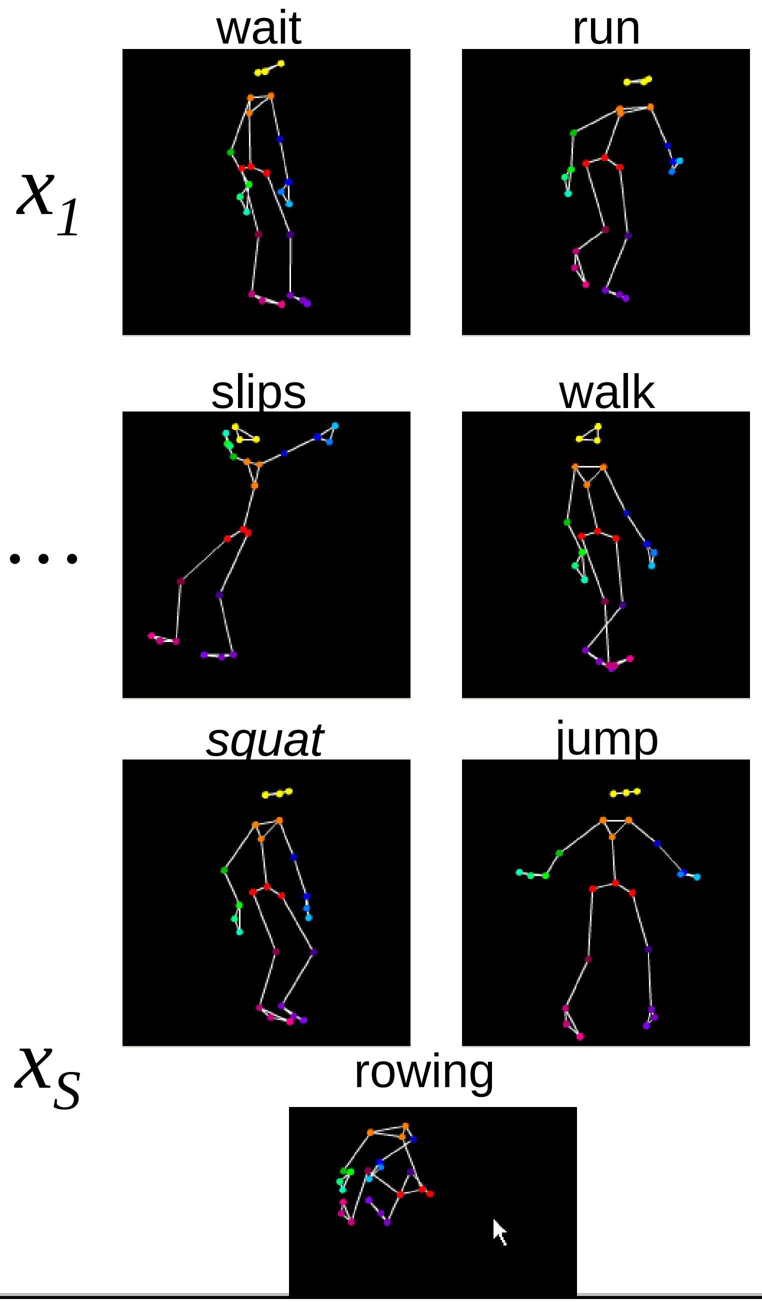
- The motion recognition process is based on the principal component analysis
- PCA decomposition of training sequences into a set of characteristic feature data, ***eigensequences*** (principal components of the original sequences)
- Eigensequences form the orthogonal basis of a linear subspace, called the ***sequence space***
- We can recognize an input sequence projecting it on the sequence space and comparing its position with those of known samples

Recognition

- The advantages of the proposed approach are the following:
  - – the set of recognizable action classes can be increased at will, as soon as training motions for the desired action classes are available
  - – short detail movements can be easily recognized
  - – changes of the actions performed by the subject can be immediately identified
  - – PCA allows reducing greatly the data dimension, providing for their real-time processing

PCA details

- Let Z be the number of action classes to recognize
- Let $\{x_i \mid i=1, \ldots, S\}$ be the complete set of training sequences
- PCA transforms $[x_i]$ → $[g_i]$ (matrix of characteristic vectors)



PCA details

- The covariance matrix of the training set is:

$$C_x = A \cdot A^t \qquad A = [(x_1 - \mu_x),...,(x_S - \mu_x)]$$

- The basis vectors of the training sequence space are the eigenvectors of C_x

- The dimension of a sequence can be reduced expressing its components in terms of the eigensequences e_1,..., e_k that are the eigenvectors corresponding to the largest k eigenvalues of C_x

PCA details

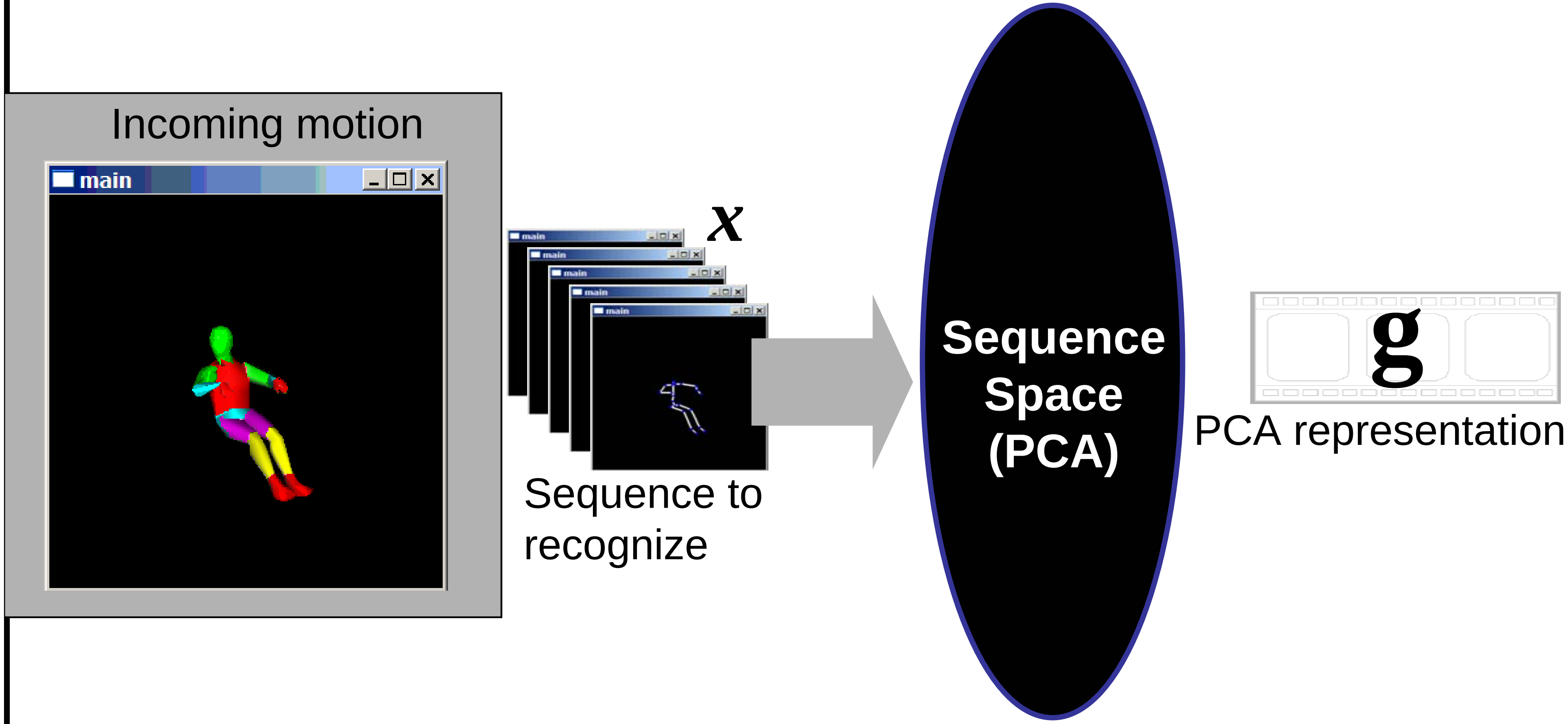
- Characteristic vector of training sequences

$$g_j = [e_1, \ldots, e_k] \cdot (x_j - \mu_x)$$

- The value of *k* corresponds to the value for which the ratio of the eigenvalue sum is above a predefined threshold ε

- An incoming sequence ***x*** is projected onto the sequence space, obtaining its characteristic vector **g**

Recognition



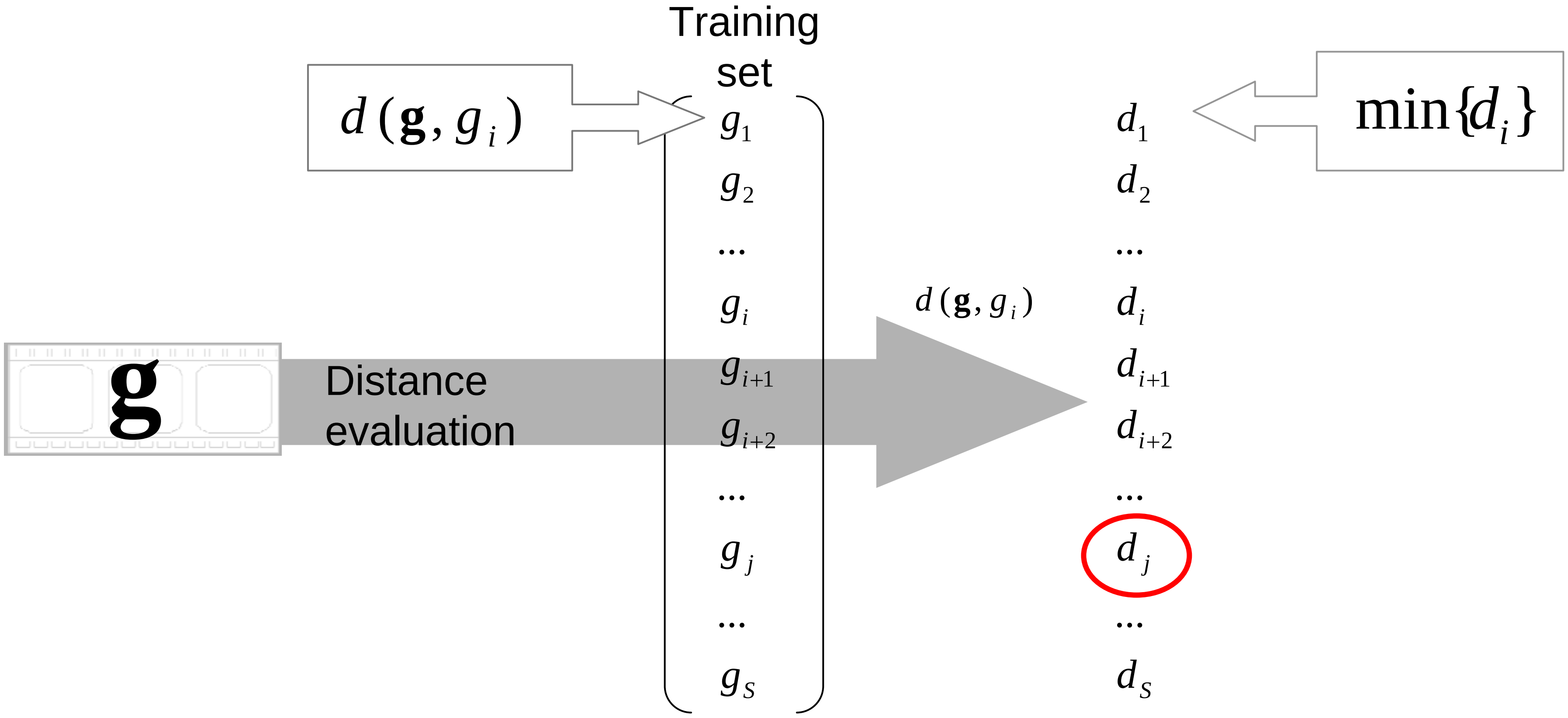
Incoming motion

Sequence to recognize

Sequence Space (PCA)

PCA representation

Recognition



Training set

$d(\mathbf{g}, g_i)$

$\min\{d_i\}$

g_1, g_2, …, g_i, g_{i+1}, g_{i+2}, …, g_j, …, g_S

$\mathbf{g}$

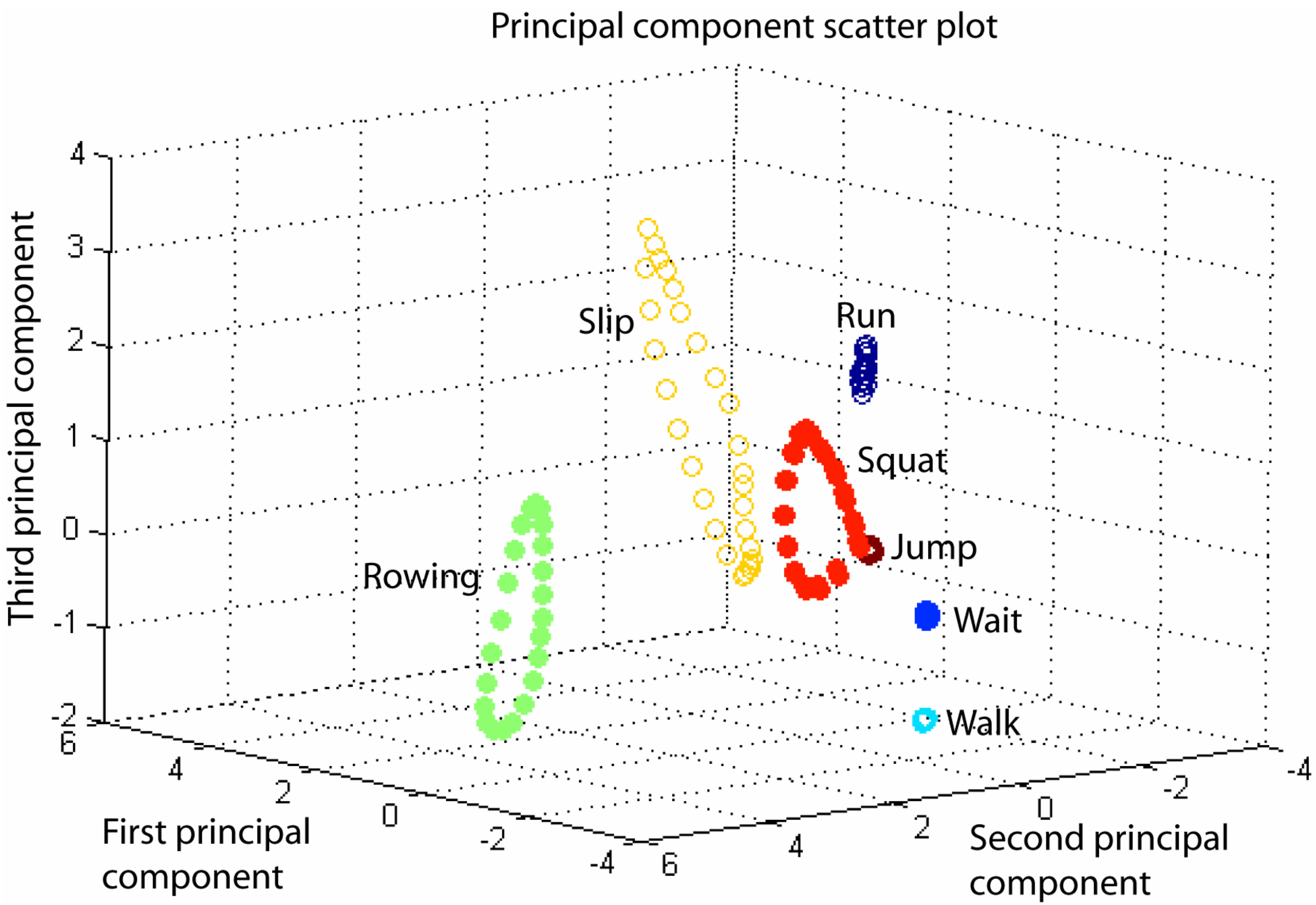
Distance evaluation

$d(\mathbf{g}, g_i)$

d_1, d_2, …, d_i, d_{i+1}, d_{i+2}, …, d_j, …, d_S

$d_j \longrightarrow$ Action class of sample *j*

Eigensequences representation

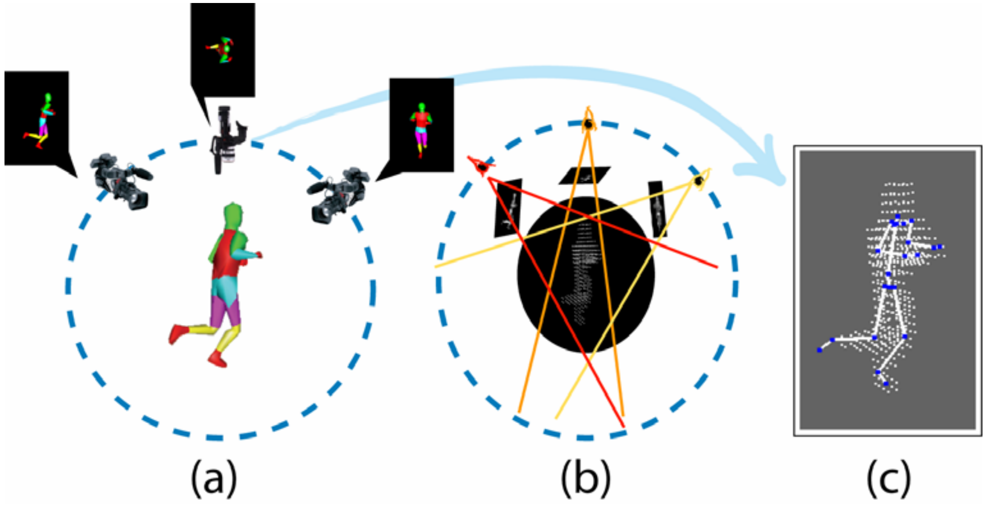


Experimental results

- Several experiments to study the behaviour of the proposed approach with respect to the different parameters involved:
  - The number *n* of frames composing a sequence
  - The distance *t* in frames between two consecutive sequences in a training motion
  - The value of the threshold ε used to select the *k* eigensequences defining the sequence space
  - The number *S* of training sequences
  - The representative motions used to extract the training sequences for each action class

Experimental results

- Experimental work in two phases.
  1. approach tested directly on the available motion capture data, taken from Motek's StockMove™GENERIC (studying the parameters)
  2. approach tested with a non-intrusive motion capture system in a virtual environment to understand how the reconstruction error introduced affects the classification process. Motion data are used to animate a dummy



Sequence dimension (*n*)

- recognition rate is substantially unaffected by the number of postures composing a sequence; this allows choosing a smaller value of *n*, providing for faster recognition of action changes
- PCA recognition applied to single postures is not effective

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f1	1	1 / 30 sec	893	0,99	7,72 msec	53,98%	46,02%
exp50-x1-f3	3	1 / 30 sec	865	0,99	7,64 msec	58,15%	41,85%
exp50-x1-f5	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%
exp50-x1-f10	10	1 / 30 sec	728	0,99	8,05 msec	87,42%	12,58%
exp50-x1-f15	15	1 / 30 sec	658	0,99	7,52 msec	87,30%	12,70%
exp50-x1-f20	20	1 / 30 sec	588	0,99	7,03 msec	87,56%	12,44%

Sequences distance (*t*)

- reducing the number of training sequences, keeping constant the number of training motions used to create them, does not affect substantially the recognition rate
- this allows to increase the value of *t*, that is to have a reduced number of sequences which are more spaced on the motion curves in the parameter spaces, reducing also the recognition time

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f5	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%
exp50-x2-f5	5	2 / 30 sec	399	0,99	4,20 msec	89,01%	10,99%
exp50-x3-f5	5	3 / 30 sec	272	0,99	2,88 msec	88,23%	11,77%
exp50-x4-f5	5	4 / 30 sec	204	0,99	2,12 msec	89,67%	10,33%

Training sequences (*S*)

- provided the same values for *n* and *t*, the recognition rates increases when the classifier uses a larger set of training motions for each action class
- increasing the number of training motions allows to describe a wider variability for the specific action

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f5-50p	5	1 / 30 sec	399	0,99	4,28 msec	77,24%	22,76%
exp50-x1-f5-75p	5	1 / 30 sec	603	0,99	6,28 msec	88,11%	11,89%
exp50-x1-f5-90p	5	1 / 30 sec	725	0,99	7,68 msec	87,93%	12,07%
exp50-x1-f5	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%

Eigenvalues threshold (ε)

- Increasing ε, the classifier uses a greater number of eigensequences, the data are represented with higher accuracy in the sequence space and the recognition error is reduced

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f5-e80	5	1 / 30 sec	798	0,8	7,56 msec	76,64%	23,36%
exp50-x1-f5-e90	5	1 / 30 sec	798	0,9	7,48 msec	84,47%	15,53%
exp50-x1-f5-e99	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%

Confusion matrix

	Out						
In	**Run**	**Wait**	**Walk**	**Rowing**	**Slips**	**Squat**	**Jumps**
Run	**96,37%**	*0,00%*	*0,00%*	*0,00%*	*0,00%*	**1,21%**	**2,42%**
Wait	*0,00%*	**100,00%**	*0,00%*	*0,00%*	*0,00%*	*0,00%*	*0,00%*
Walk	*0,00%*	*0,00%*	**75,00%**	*0,00%*	*0,00%*	*0,00%*	**25,00%**
Rowing	*0,00%*	*0,00%*	*0,00%*	**100,00%**	*0,00%*	*0,00%*	*0,00%*
Slips	*0,00%*	*0,00%*	**0,27%**	**14,52%**	**81,64%**	**2,19%**	**1,37%**
Squat	*0,00%*	*0,00%*	*0,00%*	**2,03%**	**2,54%**	**95,43%**	*0,00%*
Jumps	**0,54%**	*0,00%*	*0,00%*	*0,00%*	*0,00%*	**27,72%**	**71,74%**

Classification vs Reconstruction

- for each test set, the recognition rates using directly the motion data are compared with the recognition rates of the captured data
- the loss of quality is lower than 1% in all the cases ➔ recognition rates are relatively unaffected by the reconstruction error introduced by the non-intrusive MC system

	direct data		**reconstructed data**	
	true %	**false %**	**true %**	**false %**
exp50-x1-f5	88,41%	11,59%	88,11%	11,89%
exp50-x2-f5	88,11%	11,89%	88,89%	11,11%
exp50-x3-f5	88,23%	11,77%	87,63%	12,37%
exp50-x4-f5	89,67%	10,33%	89,31%	10,69%

Real performer

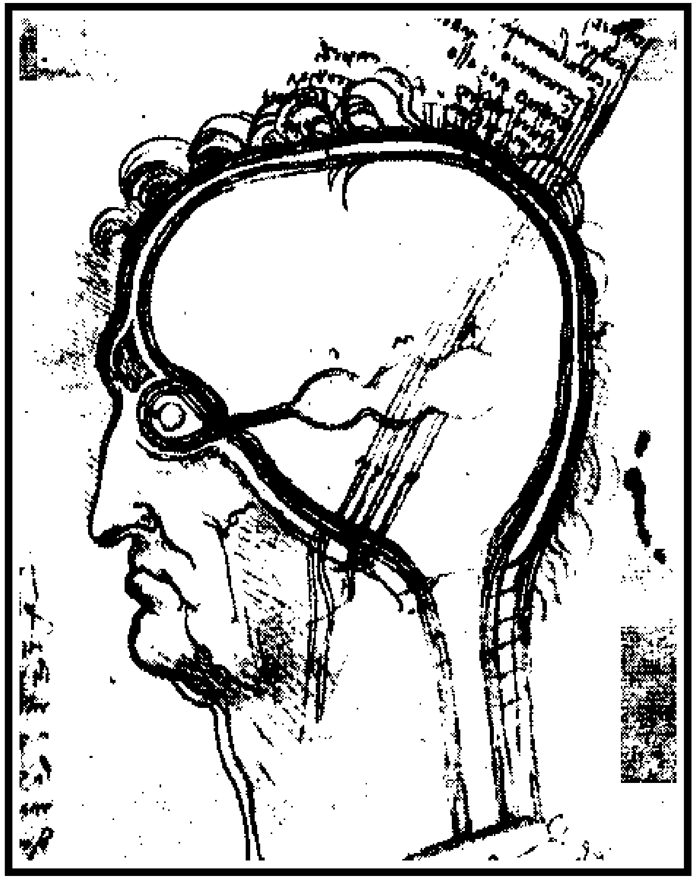
- Several actions labeled at hand
- Inter-actions frames difficult to classify even for a human observer

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
long1-x1-f5	5	1 / 30 sec	393	0,99	6,08 msec	79,41%	20,59%
long1-x1-f10	10	1 / 30 sec	353	0,99	5,97 msec	76,47%	23,53%
long1-x1-f15	15	1 / 30 sec	313	0,99	5,89 msec	75,59%	24,41%
long1-x1-f20	20	1 / 30 sec	273	0,99	5,83 msec	76,18%	23,82%
long2-x1-f5	5	1 / 30 sec	798	0,99	10,76 msec	82,65%	17,35%
long2-x1-f10	10	1 / 30 sec	728	0,99	10,76 msec	80,59%	19,41%
long2-x1-f15	15	1 / 30 sec	658	0,99	9,61 msec	79,41%	20,59%
long2-x1-f20	20	1 / 30 sec	588	0,99	9,61 msec	80,00%	20,00%

Processing times

- 2.5GHz PC with 1 GByte of RAM :
  - **Mean classification time** <11 ms
  - **Sequence space construction** (pre-processing):
    3 to 6 sec depending on the number of training sequences
  - **Motion Capture**: 30 to 40 sec per frame

Future work



- Testing the system in a real environment
- Vector comparison
  - Angle
  - Euclidean distance (weighted/unweighted)
  - Non-euclidean metrics
- Different clustering techniques (LDA, kernel PCA)
- Comparison of our method with different classification techniques (HMM, neural networks, …)

Questions?