

# Key body pose detection and movement assessment of fitness performances

by

Pablo Fernández de Dios

A Doctoral Thesis

Submitted in partial fulfilment  
of the requirements for the award of

Doctor of Philosophy  
of  
Loughborough University

April 21, 2015

© by Pablo Fernández de Dios (2015)

# Abstract

Motion segmentation plays an important role in human motion analysis. Understanding the intrinsic features of human activities represents a challenge for modern science. Current solutions usually involve computationally demanding processing and achieve the best results using expensive, intrusive motion capture devices. In this thesis, research has been carried out to develop a series of methods for affordable and effective human motion assessment in the context of stand-up physical exercises.

The objective of the research was to tackle the needs for an autonomous system that could be deployed in nursing homes or elderly people's houses, as well as rehabilitation of high profile sport performers. Firstly, it has to be designed so that instructions on physical exercises, especially in the case of elderly people, can be delivered in an understandable way. Secondly, it has to deal with the problem that some individuals may find it difficult to keep up with the programme due to physical impediments. They may also be discouraged because the activities are not stimulating or the instructions are hard to follow.

In this thesis, a series of methods for automatic assessment production, as a combination of worded feedback and motion visualisation, is presented. The methods comprise two major steps. First, a series of key body poses are identified upon a model built by a multi-class classifier from a set of frame-wise features extracted from the motion data. Second, motion alignment (or synchronisation) with a reference performance (the *tutor*) is established in order to produce a second assessment model. Numerical assessment, first, and textual feedback, after, are delivered to the user along with a 3D skeletal animation to enrich the assessment experience. This animation is produced after the demonstration of the expert is transformed to the current level of performance of the user, in order to help encourage them to engage with the programme.

The key body pose identification stage follows a two-step approach: first, the principal components of the input motion data are calculated in order to reduce the dimensionality of the input. Then, candidates of key body poses are inferred using multi-class, supervised machine learning techniques from a set of training samples. Finally, cluster analysis is used to refine the result. Key body pose identification is guaranteed to be invariant to the repetitiveness and symmetry of the performance. Results show the effectiveness of the proposed approach by comparing it against Dynamic Time Warping and Hierarchical Aligned Cluster Analysis.

The synchronisation sub-system takes advantage of the cyclic nature of the stretches that are part of the stand-up exercises subject to study in order to remove out-of-sequence identified key body poses (i.e., false positives). Two approaches are considered for performing cycle analysis: a sequential, trivial algorithm and a proposed Genetic Algorithm, with and without prior knowledge on cyclic sequence patterns. These two approaches are compared and the Genetic Algorithm

with prior knowledge shows a lower rate of false positives, but also a higher false negative rate. The GAs are also evaluated with randomly generated periodic string sequences.

The automatic assessment follows a similar approach to that of key body pose identification. A multi-class, multi-target machine learning classifier is trained with features extracted from previous motion alignment. The inferred numerical assessment levels (one per identified key body pose and involved body joint) are translated into human-understandable language via a highly-customisable, context-free grammar.

Finally, visual feedback is produced in the form of a synchronised skeletal animation of both the user's performance and the *tutor*'s. If the user's performance is well below a standard then an affine offset transformation of the skeletal motion data series to an in-between performance is performed, in order to prevent dis-encouragement from the user and still provide a reference for improvement.

At the end of this thesis, a study of the limitations of the methods in real circumstances is explored. Issues like the gimbal lock in the angular motion data, lack of accuracy of the motion capture system and the escalation of the training set are discussed. Finally, some conclusions are drawn and future work is discussed.

## Acknowledgements

I would like to first acknowledge my gratitude to my supervisors, Paul Chung and Qinggang Meng. They have always known how to (re)direct me to the right path and provide me of invaluable feedback. I have also positive words for the staff at the Computer Science: Christine, Jo, Judith, Kip, Richard, Mashhuda and many others.

To my colleagues Muhammed, Francho, Bob and many others for their support in and outside the lab. May I dedicate my best wishes for their future.

To my housemates, Jak, Johanna and Luiz for helping my first year become the best student experience possible.

To the squad of Loughborough Academicals FC. The future will surely bring a better time for them. I will always be their number one supporter.

To my bands and supporters.

To my beloved parents Paco and Mar, my main source of inspiration. Their continuous support and encouragement, as well as warm home welcoming and affection, have been decisive, as always.

To the rest of my family: my sister Alicia and my brother in-law David, my grandparents Marcial, Cari and Uja. To all, gracias.

To my sweet Heather. Thank you for every single bit. Thank you for cooking for me, for looking after the house every time I was away, for your patience and, above all, for always being there. I will always love you. Also, thanks to Perla and Brandon for making this family complete.

And finally, to my daughter, Isabelle. I will always have you in my heart. You are the driving force that makes me overcome every hurdle and reach the finish line. You are the reason for absolutely everything. I love you and I will always do.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>x</b>
<b>Glossary</b>	<b>xi</b>
<b>List of Abbreviations</b>	<b>xv</b>
<b>List of Symbols</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Overview . . . . .	2
1.3 Aim and objectives . . . . .	3
1.4 Organisation of the thesis . . . . .	6
<b>2 Review of automated assessment of fitness exercises</b>	<b>8</b>
2.1 Motion capture technologies . . . . .	8
2.2 Motion alignment methods . . . . .	9
2.2.1 DTW . . . . .	10
2.2.2 HACA . . . . .	11
2.3 Methods for approximate and exact string matching . . . . .	11
2.4 Assessment and adjustment of fitness performances given a reference . . . . .	13
2.5 Conclusions . . . . .	14
<b>3 Technical review of ad-hoc approaches to human motion assessment</b>	<b>16</b>
3.1 Related applications . . . . .	16
3.2 Analysis of approaches and identification of improvements . . . . .	23
3.3 Conclusions . . . . .	24
<b>4 System architecture</b>	<b>25</b>
4.1 Hardware configuration . . . . .	26

---

4.2	Motion data specification . . . . .	28
4.2.1	Motion sequences . . . . .	28
4.2.2	Joint motion specification . . . . .	30
4.2.3	Geometrical interpretation of motion data . . . . .	31
4.3	Conclusions . . . . .	34
<b>5</b>	<b>A method for automatic key body pose identification in motion capture data of physical exercises</b>	<b>35</b>
5.1	Overview of the method for landmark identification . . . . .	36
5.2	Motion signature calculation . . . . .	38
5.2.1	Data smoothing . . . . .	38
5.2.2	Dimensionality reduction . . . . .	39
5.3	Selection of training observations and testing observations . . . . .	40
5.4	Candidate landmark identification . . . . .	42
5.4.1	<i>Incompleteness</i> and <i>symmetry</i> of landmark sequences . . . . .	42
5.4.2	Frame-wise feature extraction . . . . .	43
5.4.3	Landmark identification . . . . .	45
5.4.4	Machine learning classifiers . . . . .	46
5.5	Landmark refining . . . . .	47
5.5.1	Landmark clustering methods . . . . .	48
5.5.2	Choosing the most suitable candidate landmark from each landmark cluster	50
5.6	Experimental results . . . . .	51
5.6.1	Evaluation measures . . . . .	52
5.6.2	Comparison of different learning and clustering techniques for landmark identification . . . . .	54
5.6.3	Comparison with Dynamic Time Warping (DTW) and Hierarchical Aligned Cluster Analysis (HACA) . . . . .	59
5.7	Conclusions . . . . .	61
<b>6</b>	<b>A genetic algorithm for periodic landmark sequence analysis based on landmark sequence fitness optimisation</b>	<b>64</b>
6.1	Theoretical background and problem formulation . . . . .	65
6.2	Analysis of periodic sequences . . . . .	68
6.2.1	Periodic analysis of a sequence . . . . .	69
6.2.2	Sequence fitness function . . . . .	70
6.2.3	A naive algorithm for extraneous member deletion on string sequences	71
6.2.4	An approach based on the Needleman-Wunsch algorithm for periodic sequences . . . . .	73
6.3	A genetic algorithm for periodic sequences analysis . . . . .	73
6.3.1	Genetic algorithms . . . . .	73
6.3.2	Fitness operators for sequences and sequence members . . . . .	75
6.3.3	Two mutation operators for sequences . . . . .	77
6.3.4	A probabilistic tournament selection operator for sequences . . . . .	78
6.3.5	A single point crossover operator for sequences . . . . .	79
6.4	Experimental results . . . . .	80
6.4.1	Evaluation on randomly generated periodic sequences . . . . .	82
6.4.2	Evaluation on periodic sequence analysis of key body pose sequences .	86
6.5	Conclusions . . . . .	87

<b>7</b>	<b>Automatic, landmark-based feedback production based on adjusted parameters</b>	<b>90</b>
7.1	Overview of the methods for landmark assessment and motion adjustment . . .	90
7.2	Automatic production of textual assessment for synchronised motion series . .	92
7.2.1	Training observations and testing observations for assessment modelling	93
7.2.2	Assessment of identified landmarks . . . . .	94
7.2.3	A parameter to model the current overall level of performance . . . . .	95
7.2.4	Feedback generation from landmark assessment: an assessment <i>meta</i> - grammar and an assessment tree . . . . .	96
7.3	Motion adjustment of natural demonstration for motivational research . . . . .	103
7.3.1	An algorithm to synchronise two incomplete and asymmetric landmark sequences . . . . .	104
7.3.2	A method to synchronise two motion sequences, given their landmark sequences . . . . .	104
7.3.3	An affine offset transformation of a user to a tutor . . . . .	105
7.4	Experimental results . . . . .	106
7.4.1	Landmark assessment . . . . .	107
7.4.2	Generated assessment feedback . . . . .	111
7.5	Conclusions . . . . .	113
<b>8</b>	<b>Limitations of the approach and technical issues</b>	<b>114</b>
8.1	Accuracy of the Kinect camera: self-occlusion and orientation of the individual	114
8.2	Discontinuities in the Euler angles due to gimbal lock . . . . .	115
8.2.1	Unsuitable motion signature . . . . .	117
8.2.2	Unnatural adjustment reconstruction . . . . .	118
8.3	Escalation of training samples . . . . .	119
8.4	Conclusions . . . . .	120
<b>9</b>	<b>Conclusions and future work</b>	<b>122</b>
9.1	Contributions . . . . .	123
9.2	Analysis of issues and strengths . . . . .	125
9.3	Future work . . . . .	125
	<b>List of Publications</b>	<b>127</b>
	<b>List of References</b>	<b>128</b>
<b>A</b>	<b>Results for landmark identification and landmark refining</b>	<b>140</b>
<b>B</b>	<b>Results for periodic landmark sequence analysis and landmark assessment</b>	<b>153</b>
<b>C</b>	<b>Comparison of results for landmark identification and landmark refining with DTW</b>	<b>174</b>
<b>D</b>	<b>Generated landmark assessment, motion synchronisation and motion adjustment</b>	<b>189</b>
<b>E</b>	<b>Grammars for assessment</b>	<b>194</b>

# List of Figures

1.1	Diagram showing the basic setup for exercise assessment . . . . .	4
1.2	“Stick man” representation of each motion class . . . . .	5
1.3	Key body poses trained by the system . . . . .	5
2.1	Example of segmentation of HACA with motion data samples from the CMU Motion Capture (MoCap) database . . . . .	12
4.1	Diagram representation of the system . . . . .	25
4.2	Biovision Hierarchy (BVH) file format joints hierarchy . . . . .	27
4.3	Skeleton depicted with the reference and local coordinate systems . . . . .	32
5.1	Skeletal and plot representation of an instance of the <i>Ankles</i> exercise . . . . .	36
5.2	Key body pose learning and identification subsystem . . . . .	37
5.3	Motion signature calculation diagram . . . . .	37
5.4	Pre-processing diagram . . . . .	41
5.5	Example of extracted frame-wise features . . . . .	43
5.6	Illustration of landmark identification and landmark refining stages . . . . .	49
5.7	Example of compromise intervals calculated automatically . . . . .	54
5.8	Comparison of results for landmark identification and landmark refining of C4.5 and support vector machines . . . . .	57
5.9	Comparison of average precision achieved for landmark identification+landmark refining per testing sample using/not using the motion signature and extracting/not extracting frame-wise features . . . . .	58
5.10	Representation of results for landmark identification and landmark refining . . . . .	60
5.11	Comparison of results between landmark identification+landmark refining and DTW . . . . .	62
6.1	Charts showing performance of the Genetic Algorithm (GA) and the $GA^K$ for different number of generations, with a population size of 40 and a mutation probability of 0.7 . . . . .	83
6.2	Chart showing performance of the GA and the $GA^K$ for different population sizes, with 40 generations and a mutation probability of 0.7 . . . . .	84
6.3	Chart showing performance of the GA and the $GA^K$ for different mutation probabilities, with 40 generations and 10 individuals per generation . . . . .	84
6.4	Chart showing a comparison between performance achieved by string matching algorithms and the GAs with periodic sequences with different rate of in-sequence extraneous members. GAs run in 50 generations, 10 individuals per generation and a mutation probability of 0.3 . . . . .	85
6.5	Comparison of results for periodic landmark sequence analysis between the trivial algorithm and the GA . . . . .	88

7.1	System diagram depicting the periodic landmark sequence analysis, landmark assessment and motion adjustment stages . . . . .	91
7.2	System diagram depicting the landmark assessment stage . . . . .	93
7.3	Class diagram for the assessment tree . . . . .	98
7.4	Assessment tree produced after ANTLR parsing of Listing 7.3 . . . . .	101
7.5	Assessment tree produced after ANTLR parsing of Listing 7.3 - <i>exp7</i> (continued from Figure 7.4) . . . . .	102
7.6	System diagram depicting tutor's motion data motion synchronisation and motion adjustment stages . . . . .	103
7.7	Representation of results for periodic landmark sequence analysis and landmark assessment . . . . .	109
7.8	Generated assessment and adjustment for <i>p00_arms_worse03</i> . . . . .	111
7.9	Generated assessment and adjustment for <i>p04_inner_thighs_worse</i> . . . . .	112
8.1	Animation of a motion capture error . . . . .	115
8.2	Euler angles of a failed recovered sample . . . . .	115
8.3	Illustration of gimbal lock in motion adjustment for the <i>p01_arms_normal01</i> sample . . . . .	116
8.4	Comparison of the calculated motion signature between absolute joint positions and Euler angles of the <i>p01_shoulders_normal01</i> sample . . . . .	117
8.5	Absolute positions and Euler angles of the <i>p01_shoulders_normal01</i> sample . . . . .	118
8.6	Illustration of gimbal lock in motion adjustment (II) . . . . .	119
A.1	Representation of results for landmark identification and landmark refining in the <i>Ankles</i> exercise . . . . .	140
A.2	Representation of results for landmark identification and landmark refining in the <i>Arms</i> exercise . . . . .	143
A.3	Representation of results for landmark identification and landmark refining in the <i>Calves</i> exercise . . . . .	146
A.4	Representation of results for landmark identification and landmark refining in the <i>Inner thighs</i> exercise . . . . .	148
A.5	Representation of results for landmark identification and landmark refining in the <i>Shoulders</i> exercise . . . . .	151
B.1	Representation of results for periodic landmark sequence analysis and landmark assessment in the <i>Ankles</i> exercise . . . . .	153
B.2	Representation of results for periodic landmark sequence analysis and landmark assessment in the <i>Arms</i> exercise . . . . .	170
B.3	Representation of results for periodic landmark sequence analysis and landmark assessment in the <i>Calves</i> exercise . . . . .	171
B.4	Representation of results for periodic landmark sequence analysis and landmark assessment in the <i>Inner thighs</i> exercise . . . . .	172
B.5	Representation of results for periodic landmark sequence analysis and landmark assessment in the <i>Shoulders</i> exercise . . . . .	173
C.1	Comparison of results between landmark identification+landmark refining and DTW in the <i>Ankles</i> exercise . . . . .	174
C.2	Comparison of results between landmark identification+landmark refining and DTW in the <i>Arms</i> exercise . . . . .	177

---

C.3	Comparison of results between landmark identification+landmark refining and DTW in the <i>Calves</i> exercise . . . . .	180
C.4	Comparison of results between landmark identification+landmark refining and DTW in the <i>Inner thighs</i> exercise . . . . .	183
C.5	Comparison of results between landmark identification+landmark refining and DTW in the <i>Shoulders</i> exercise . . . . .	186
D.1	Generated assessment and adjustment for <i>p00_arms_worse03</i> . . . . .	190
D.2	Generated assessment and adjustment for <i>p01_shoulders_worse</i> . . . . .	191
D.3	Generated assessment and adjustment for <i>p04_inner_thighs_worse</i> . . . . .	192
D.4	Generated assessment and adjustment for <i>p06_ankles_worse</i> . . . . .	193

# List of Tables

3.1	Summary of reviewed motion capture and tracking approaches . . . . .	17
5.1	Details of the features of the training and testing samples for landmark identification and landmark refining . . . . .	51
5.2	Involved body joints per exercise . . . . .	51
5.3	Bio-information of the participants . . . . .	52
5.4	Comparison of precision of performance of different machine learning classifiers in landmark identification . . . . .	55
5.5	Comparison of precision and accuracy for landmark identification+landmark refining using different algorithms . . . . .	56
5.6	Comparison of execution time for landmark refining using different algorithms . . . . .	58
5.7	Average proficiency achieved for landmark identification and landmark refining . . . . .	59
5.8	Comparison of proficiency for landmark identification between the proposed approach, DTW and HACA . . . . .	59
5.9	Average execution time for landmark identification using several different methods . . . . .	61
6.1	Comparison of accuracy ( $a$ ), recall ( $r$ ), true negative rate ( $tnr$ ), false positive rate ( $fpr$ ), false discovery rate ( $fdr$ ) and average execution time ( $ET$ ) (in $s$ ) achieved using several different methods for random, periodic sequence analysis. GAs run over 40 generations of 10 individuals each and a mutation probability of 0.3 . . . . .	85
6.2	Comparison of true negative rate ( $tnr$ ), false positive rate ( $fpr$ ), false discovery rate ( $fdr$ ), false negative rate ( $fnr$ ) and average execution time per testing sample ( $ET$ ) (in $s$ ) of identified key body poses achieved on periodic sequence analysis using the naive algorithm . . . . .	86
6.3	Comparison of true negative rate ( $tnr$ ), false positive rate ( $fpr$ ), false discovery rate ( $fdr$ ), false negative rate ( $fnr$ ) and average execution time per testing sample ( $ET$ ) (in $s$ ) of identified key body poses achieved on periodic sequence analysis using both GAs with ( $GA^K$ ) and without ( $GA$ ) prior knowledge . . . . .	86
7.1	Details of parameters of the training samples and testing samples for landmark assessment per exercise . . . . .	107
7.2	Average proficiency achieved on landmark assessment . . . . .	108

# Glossary

**AdaBoost** Machine learning meta-algorithm based on modelling a '*stronger learner*' upon combination of '*weaker learners*'.

**assessed landmark** Identified landmark after landmark assessment.

**assessment level** Numerical value of the landmark assessment of a specific body joint on a specific landmark (of a specific landmark type).

**body joint** A specific joint of the human body, e.g., shoulder, elbow, knee, hip,....

**C4.5** Machine learning classifier based on a decision tree algorithm.

**candidate landmark** An identified landmark made by the machine learning classifier upon motion data of a testing sample and before landmark refining.

**compromise interval** Range of frames within the motion sequence, associated with a ground truth landmark, to evaluate the precision of the landmark identification and periodic landmark sequence analysis stages.

**feature** Attribute values of an observation, upon which classification is performed.

**frame-wise feature** Feature of an observation conforming to a noise-modelling computation method of the motion signature.

**frame-wise features observation** An observation made up of frame-wise features calculated from the motion signature.

**ground truth assessment** Assessment levels of a ground truth landmark (or all of the ground truth landmarks) of a given performance.

**ground truth landmark** Key body pose or landmark of an individual's performance, labelled with ground truth data, e.g. frame, landmark type,....

**identified landmark** A key body pose identified from the motion data in one of the stages of the method.

**landmark** Key body pose, of a specific landmark type.

**landmark assessment** Action of inferring the execution level of a performance (i.e., the quality of the user's performance of an exercise's motion class) by inferring a series of assessment levels on each identified landmark.

**landmark cluster** A set of candidate landmarks grouped in the landmark refining stage according to a specific method.

**landmark cluster selection** A method for selecting one candidate landmark from each of a set of landmark clusters.

**landmark identification** Action of extracting a series of landmarks from a motion sequence (a testing sample) using a model built from training samples.

**landmark model training** Stage in which a model to classify landmarks, upon the values of their features, is built. The training observations are used to train the machine learning classifier.

**landmark period analysis** Analysis of the periodic nature of a landmark sequence.

**landmark refining** Improvement of a candidate landmark by means of cluster analysis.

**landmark sequence fitness** *Fitness* (or aptitude test outcome) of a landmark sequence, calculated from all the landmark sequence member fitnesses within it.

**landmark sequence member fitness** *Fitness* (or aptitude test outcome) of a landmark within a landmark sequence.

**landmark sequence period fitness** *Fitness* (or aptitude test outcome) of an identified landmark type period, within a landmark sequence.

**landmark sequence repetitiveness** Characteristic of a given sequence of key body poses, within a performance, in which the stretches involved in the exercise are repeated a number of times.

**landmark sequence symmetry** Characteristic of a given sequence of key body poses, within a performance, in which the limbs of one and the opposite side of the body are executed alternately.

**landmark sequence type cycle** A cyclic pattern found within a landmark sequence, attending to the value of their landmark types.

**landmark type** Kind of key body pose of landmark within an average performance of an exercise.

**machine learning classifier** Statistical, machine learning-based method to infer a series of observation classes over a set of input observations, representing a range of real data.

**motion adjustment** Transformation of the data of a performance in which modification of the appearance of a number of poses into that of a reference is performed, by preserving original values like complexion, scale and orientation.

**motion alignment** Action of matching or pairing a number of time frames of several motion sequences in order to transform or compare them.

**motion class** Class of exercise to which a given performance belongs.

**motion sequence** A series of data describing a sample human motion (e.g. values of the body joint angles, absolute positions,...) in time.

**motion signature** A dimension-reduced motion data series of a motion sequence.

**motion synchronisation** Motion alignment consisting of matching two sequences of key body poses or landmarks, of two motion sequences, given their landmark types.

**multi-class** Type of machine learning classifier that takes more than two values for the classified observation class.

**multi-target** Feature of a machine learning classifier that infers the value of several observation classes on each input observation.

**Naive Bayes** Probabilistic machine learning classifier based on the application of Bayes' theorem to independent features.

**observation** A set of features from the input sample to be classified based on its values, representing a time frame.

**observation class** Attribute or feature of an observation to be classified, i.e., inferred by the machine learning classifier.

**performance** An execution of an exercise, of a given motion class, by an individual.

**performance level parameter** Value representing the current level of performance of a user and used to perform tailored motion adjustment.

**periodic landmark sequence analysis** Stage of motion synchronisation in which out-of-sequence landmarks are removed from a given set  $\bar{\chi}$  of refined landmarks, which contains periodic sequences of landmarks.

**range of motion** Limits of a stretch, within which a subject can move one of their limbs freely, without incurring physiological pain.

**refined landmark** A landmark extracted from a set of clusters after landmark refining.

**testing observation** One of the observations of a given testing sample.

**testing sample** A performances object of landmark identification and landmark assessment.

**training observation** One of the observations of a given training sample.

**training sample** One of the performances chosen to train the system (i.e., build the machine learning classifier's model).

**training set** Performances chosen to train the system (i.e., build the machine learning classifier's model).

**tutor** Reference performance of a give motion class, i.e., an exemplary performance of an exercise that is normally shown to the user as a demonstration of such.

**user** Individual performing an exercise, normally to be evaluated.

# List of Abbreviations

**2D** two-dimensional.

**3D** three-dimensional.

**ADLs** Activities of daily living.

**BVH** Biovision Hierarchy.

**CMOS** Complementary Metal-Oxide Semiconductor.

**DOF** Degree of Freedom.

**DR** Dimensionality Reduction.

**DTW** Dynamic Time Warping.

**GA** Genetic Algorithm.

**H-C** Human-Computer.

**HACA** Hierarchical Aligned Cluster Analysis.

**IK** Inverse Kinematics.

**IR** infrared.

**ITC-SOPI** ITC-Sense Of Presence Inventory.

**MoCap** Motion Capture.

**PC** Principal Component.

**PCA** Principal Component Analysis.

**POI** Point Of Interest.

**RAM** Random-Access Memory.

**RGB** Red-Green-Blue.

**RoM** range of motion.

**RT** real-time.

**RVT** Reactive Virtual Trainer.

**SMO** Sequential Minimal Optimisation.

**SVD** Singular Value Decomposition.

**SVM** support vector machines.

.

**ToF** Time-of-Flight.

# List of Symbols

$D$  Product of the number of involved body joints ( $N_J$ ) and the number of involved Degrees Of Freedom (DOFs) ( $N_{DOF}$ ).

$N_A$  Number of possible assessment values that the observation classes of an landmark assessment observation  $\mathcal{O}$  can take, i.e., in the range  $0..N_A$ .

$N_{DOF}$  Number of axis coordinates per body joint. For instance, angles and positions have 3, while quaternions have 4.

$N_J$  Number of body joints selected for analysis (i.e.,  $|J|$ ).

$N_{Tr}$  Number of time frames of the training sample (i.e.,  $|Tr|$ ).

$N_{Ts}$  Number of time frames of the testing sample (i.e.,  $|Ts|$ ).

$N_T$  Number of time frames of the motion data (i.e.,  $|T|$ ).

$N_\sigma$  Number of landmark clusters generated in the landmark refining stage.

$T_c$  Algorithm to perform landmark clustering based on the time separation threshold parameter  $\beta$ .

$Y$  Domain of the observation class attribute.

$\Delta_k$  Number of Principal Components (PCs) which accumulated level of representation  $\rho$  is bigger than a given value.

$\Delta_\Lambda$  Compromise interval threshold.

$\Gamma$  Function of the past values of the performance level parameter ( $\gamma_{j_i}$ ) that produces the current value for  $\gamma$ .

$\Lambda$  One of the ends of a compromise interval.

- $\Phi$  General purpose machine learning classifier.
- $\Psi$  General purpose sub-classifier (i.e., a machine learning classifier that is part of a higher-level classifier).
- $\alpha$  Accuracy of a landmark identification or motion synchronisation evaluation.
- $\beta$  Magnitude used to measure the landmark cluster threshold separation in time (in *frames*).
- $\chi^G$  Key body poses or landmarks of an individual's performance, labelled with ground truth data, e.g. frame, landmark type,....
- $\chi$  Sequence of landmarks or key body poses.
- $\gamma$  Current level of performance (performance level parameter) of the user, from the perspective of previous performances.
- $\mathcal{O}^{FWF}$  frame-wise features observation of a testing sample.
- $\mathcal{O}^{Tr}$  Observations of the training samples.
- $\mathcal{O}^{Ts}$  Observations of the testing samples.
- $\mathbb{T}r$  Set of integer values denoting the time frames of a training sample that will be used to train the landmark identification and landmark assessment subsystems.
- $\mathbb{T}s$  Set of integer values denoting the time frames that a testing sample is being tested for landmark identification and landmark assessment.
- $\kappa$  The index of the first appearance, after a given index, in a sequence of a given landmark type.
- $\mathbb{A}$  A multi-class, multi-target machine learning classifier of observations that infers a value of their observation class attributes from the values of their features.
- $\mathbb{F}$  A multi-class machine learning classifier of observations that infers a value of its observation class attribute upon the values of their features.
- $\mathbb{S}$  A motion synchronisation between two motion sequences (or sub-sequences).
- $\mathbb{T}$  Set of integer values denoting the time frames that the performance's motion data represents.
- $\mathcal{A}^*$  All possible sequences defined by the alphabet  $\mathcal{A}$ .
- $\mathcal{A}$  An alphabet of string characters.

- $\mathcal{M}$  Function to retrieve all the landmark sequence type cycles contained on a landmark (or landmark type) sequence.
- $\mathcal{N}$  All the landmark sequence type cycles contained in a landmark (or landmark type) sequence.
- $\mathcal{P}$  A series of training pattern periods.
- $\mathfrak{F}$  Function to calculate the fitness of a landmark (landmark sequence member fitness) within a sequence or that of the whole sequence (landmark sequence fitness).
- $\mathfrak{M}$  Function class denoting any function of time and body joint on motion data (e.g., angles, absolute positions, ...).
- $\mathcal{O}$  Observation of a sample, with a set of features  $x$  and observation classes  $y$ .
- $\mathcal{C}$  Motion re-sampling function.
- $\mathcal{J}$  Set of body joints selected for analysis and feature extraction in the motion class.
- $j$  A body joint, so that  $j \in \mathcal{J}$ .
- $\mathfrak{s}$  A pair of synchronised landmarks belonging to different landmark sequences (of two different performances).
- $\mu$  A landmark cluster selection algorithm.
- $\nu$  A landmark sequence type cycle.
- $\omega$  Motion re-sampling time window (in  $s$ ).
- $\bar{\chi}^S$  Result of a periodic landmark sequence analysis over  $\bar{\chi}$ .
- $\bar{\chi}$  Result of landmark refining over  $\chi$ .
- $\bar{\mathcal{P}}$  Function to calculate the mean body pose values of a motion sequence per landmark type.
- $\bar{\bar{\chi}}$  Result of landmark assessment of  $\bar{\chi}^S$ .
- $\phi$  Euler angle *heading* or rotation about the OW axis.
- $\pi$  Precision of a landmark identification or motion synchronisation evaluation.
- $\psi$  Euler angle *bank* or rotation about the OV axis.
- $\rho$  Level of representation of a PC.

$\sigma$  A landmark cluster generated in the landmark refining stage.

$\tau$  The value of a landmark type.

$\mathbf{AL}(\chi^G)$  Assessment levels of the ground truth landmarks ( $\chi^G$ ).

$\mathbf{AL}(\bar{\chi})$  Assessment levels of the assessed landmarks ( $\bar{\chi}$ ).

$\theta$  Euler angle *attitude* or rotation about the OU axis.

$t$  A motion's time frame, so that  $t \in \mathbb{T}$ .

$x$  Representation of a feature of an observation.

$y$  Representation of a observation class of an observation.

*In loving memory of my grandmother, Uja. Descanse en paz.*

# Chapter 1

## Introduction

### 1.1 Motivation

Whilst modern medical science is contributing towards early detection and treatment of diseases related with the age, most of the results involve medication and clinical trials. The later have demonstrated the benefits of both cognitive and physical exercising for people in a stage of life where the risk of contracting age-related diseases is high. While the desired approach is to follow a program of activities, regularly supervised by a professional trainer, the majority of elderly individuals cannot afford that. Moreover, there is a lack of specialists who could satisfy the demand in nursing homes, hospitals or patients homes.

Therefore, the aim of this research was to design an autonomous, intelligent system that would meet several important requirements. Firstly, it has to be designed so that instructions on physical exercises, specifically directed to elderly people, are delivered in an understandable way. Secondly, it has to deal with subjects who cannot necessarily perform all of the prescribed movements. This may be due to physical impediments, lack of motivation or difficulty in following the movements.

The association between physical exercise on a daily basis and the prevention of age-related illnesses has not been completely established by modern science. Whether one is a direct cause of the other is a question that still remains open [77, 119]. Nevertheless, current results point to a belief that elderly individuals involved in more and regular physical activities have a lower risk of contracting cardiovascular and Alzheimer-related diseases [26, 35, 64, 124].

Additionally, some authors defend the participation of elderly people on exergames as a way of reducing the risk of contracting dementia-related diseases [115], preventing falls [100], balance improvement [114] and help in rehabilitation [15]. Hence the need for an autonomous, intelligent system that meets the important requirements.

Given the above and the fact that the elderly population is growing, a system that can help them to be engaged with physical activity is needed. Most people cannot afford a personal trainer. Some are unable to move from their home/residence. Finally, modern equipment and methodologies are not adapted to the requirements of elderly people.

## 1.2 Overview

This thesis proposes an approach to automatic assessment of physical exercises based on a methodology for automatic, semi-supervised stand-up exercises assessment given a Motion Capture (MoCap) of body joints data stream. The methodology comprises a two-step procedure. First, key body poses (or landmarks, sequence of which expressed as  $\chi$ ) are identified and labelled. Then, each identified landmark is given a score for each involved body joint, based on a performance comparison with a reference (the tutor). These scores are collected and human-understandable feedback is generated based on the scores.

For the first step, a human motion segmentation and alignment framework (consisting of landmark identification and landmark refining), based on the application of a multi-class learning algorithm and cluster analysis, is presented. The addressed problem is to identify a series of key time momenta (landmarks) on each kind of motion class (exercise) that are common for each member of the same exercise class. The key contribution of this work is the specification of a method to train a machine learning classifier in order to identify key body poses from MoCap data, given a set of frames labelled by an expert as ground truth (the ground truth landmarks, expressed as  $\chi^G$ ) with a high level of accuracy and completeness. The method applies a multi-class machine learning classifier on dimensionality-reduced motion data and performs very well against variations on the performance in terms of landmark sequence symmetry and landmark sequence repetitiveness. It is a systematic framework for motion matching, a key step towards performance assessment.

The second step involves first key body poses sequence analysis (or periodic landmark sequence analysis) in order to remove inconsistencies in the previously identified landmarks that could

prevent correct motion alignment with the reference to be assessed against. Then, landmark assessment of the identified landmarks is done by means of a multi-class, multi-target machine learning classifier operating on features calculated upon motion synchronisation. The result is a series of numerical values or scores for each identified landmark and body joint that is then translated into human-understandable language. To enrich the capabilities of the verbal assessment, visual assessment is produced by a context-free grammar based on the values of the scores for each body joint.

The background of this project has been previously reviewed, to different extents, in the literature. Assisting systems in home environment applications, like measurement of user's immersion and motivation, posture of back and shoulders and supervised physical activities, are typical examples of research lines showing promising results. Given the nature of the current envisioned project, human body analysis methodologies deserves special attention. Dancing is one of the fields where real-time human motion tracking has made some advancements.

### **1.3 Aim and objectives**

The aim of this thesis is to design an innovative real-time, adaptable, interactive system that assesses elderly people performing simple exercises. Preliminary review of the current state of the art suggests a series of research issues that have not been successfully tackled by existing solutions. These include:

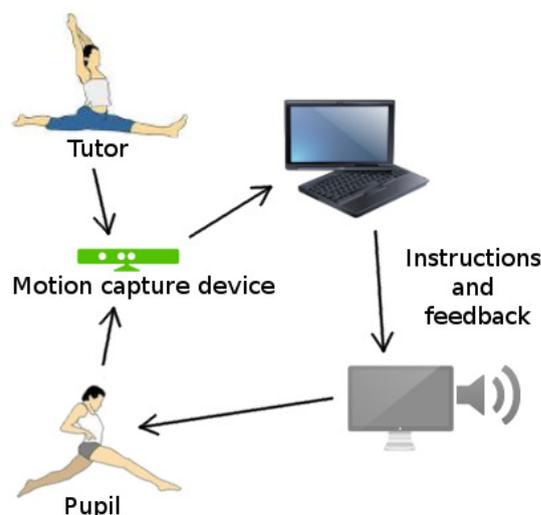
- Reliable, non-intrusive motion capture of a human in an uncontrolled scenario
- Autonomous motion analysis and matching with ground truth data
- Realistic assessment of physical exercise performances
- Real-time, unsupervised feedback and posture correction
- Keep history data and use that information to provide further, customised feedback, e.g. from a health specialist or a physician

The above requirements have been herein partially addressed due to the broad complexity of the field. These are, in the short run:

- **Using off-the-shelf, non-intrusive human motion tracking technology and experiment and report on its usability and accuracy**
- **Develop on-line preprocessing and processing procedures that can be implemented in real-time**
- **Produce human-understandable feedback and assessment on a specific set of stand-up exercises (with views on a most probable escalation of the domain), in a semi-supervised fashion**
- **Conduct a series of motion data recordings of people performing the addressed exercises, in order to support semi-supervised motion data analysis methods**
- **Design a proof of concept to convey the above, bringing the real-time requirement to a second level**

The setting-up of the envisaged scenario for exercise assessment has been designed as shown in Figure 1.1. First, the system is trained off-line with the performances of a number of individuals, under various conditions and with different levels of performance. This includes the demonstration of an exemplary or reference performance (tutor). All these training samples are recorded with a MoCap device (namely, a Kinect camera). Then, the individual to be assessed, or user, proceeds to do the instructed exercise. Their performance is also recorded and processed. Finally, visual and natural language feedback is delivered by means of video and audio devices, respectively.

FIGURE 1.1: Diagram showing the basic setup for exercise assessment



The selected exercises are illustrated in Figure 1.2 and Figure 1.3. These exercises involve stretching of a number of joints of both upper and lower extremities. They were picked from the “ElderGym” self-guide available at [92].

FIGURE 1.2: “Stick man” representation of each motion class

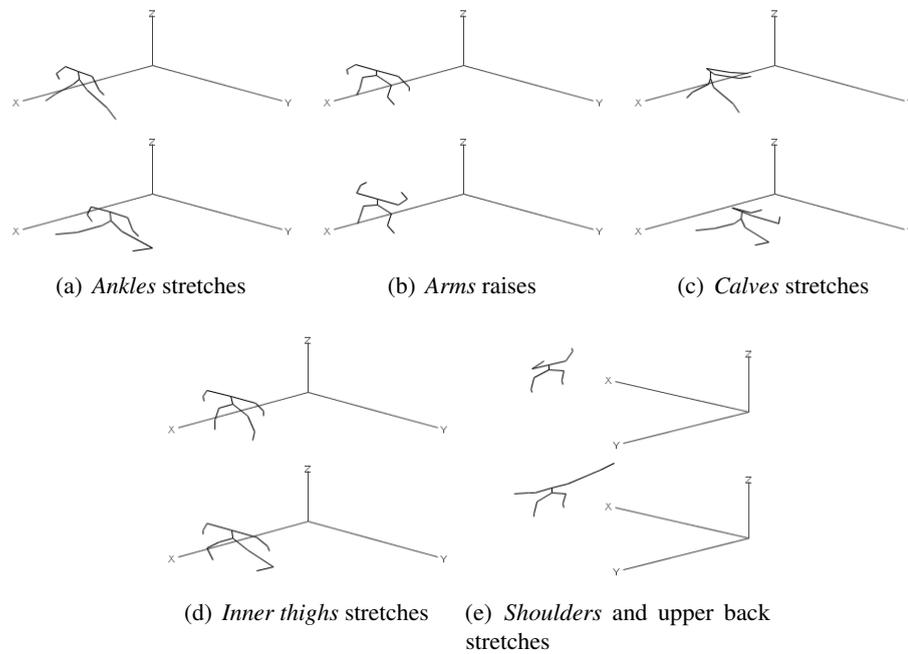


FIGURE 1.3: Key body poses trained by the system. Each landmark type is associated with a range of motion (RoM) edge within the performance



One key objective is to tackle the problem of accurately aligning two or more sequences of human motion data. This is important for an assessment to be reliably carried out by comparing a reference motion (tutor) of the performance of an exercise with the one produced by the individual to be assessed (user).

Furthermore, the way the assessment itself is performed and then delivered to the user should also be worked out. A metric for motion data comparison should be designated and assessed with the test bed data. Most existing systems implement a virtual trainer that encourages the user to perform better, or simply show a series of numbers representing the difference between the performance and the reference. These methods need to be expanded to bring the human-machine interaction to a different level of immersion and understanding. That is, the system should be able to deliver specific and understandable information to the user, who may not be familiar with technical language or details.

Finally, it is understood that the task of encouraging elderly people in physical exercise may not be complete if the virtual trainer is not able to adapt to the personal circumstances of the individual. The instructions that help the user understanding the activities to be performed should take into account their latest achievements and adjust the level of difficulty to their condition.

## **1.4 Organisation of the thesis**

The structure of this thesis is as follows. First, a review of existing methods for assessment of exercises is analysed in Chapter 2. Various MoCap technologies are discussed, as well as different methods for motion alignment. This field of research is strongly related with the approach presented in this thesis. Finally, the reviewed methods for exercise assessment and adaptation are presented. In Chapter 3, a technical review of the related applications is reviewed. A summary of the improvements that are tackled by the proposed approach is also provided.

The architecture of the proposed system is presented in Chapter 4, where the different methods for landmark identification and landmark assessment are described. Then, the hardware configuration of the MoCap sub-system and the format of the MoCap data are briefly commented. The standards followed to define the motion data are then explained, as well as its geometrical interpretation. Finally, some technological aspects of the MoCap architecture and other dimension reduction and motion alignment algorithms are summarised.

In Chapter 5, the method for automatic landmark identification is presented and evaluated. The sub-processes are explained in detail, including Dimensionality Reduction (DR), data smoothing, feature calculation, system training and landmark refining. The accuracy of the landmark identification method is then evaluated by cross-validation with the ground truth. A Genetic Algorithm (GA) for out-of-sequence members removal is discussed in Chapter 6. The aim of

---

this algorithm is to remove false positives (i.e., wrongly identified body poses) from periodic sequences. The GA is benchmarked with string matching approaches. Chapter 7 outlines a series of methods for motion synchronisation, automatic landmark assessment and motion adjustment, followed by an analysis of these methods. The assessment prediction is evaluated by cross-validation with the ground truth and the generated feedback of a number of samples are presented, as well as the skeletal reconstruction of the adjusted performances. Finally, the limitations of the methods are discussed in Chapter 8 and the conclusions, key contributions and future work can be found in Chapter 9.

## Chapter 2

# Review of automated assessment of fitness exercises

Several studies developed in the last decades have dedicated time and human resources on “ageing well”. This area has experienced a rising interest in modern society and has been sponsored by, among others, global research institutions in the European Union and the United States of America. The quality of living in the later stages of life is receiving global awareness, responding to the accomplishment of a higher priority in modern societies. Science plays an invaluable role towards these changes by contributing the latest advances in intelligent, household technologies that help independent living for elderly people.

Motion Capture (MoCap) technologies bring a real-time, reliable means of gathering data of human motion. In order to analyse the performance of the individuals, their movements have to be first captured and then transformed into a manageable format. Previous research on MoCap, *motion matching and alignment* and *motion assessment* are summarised and analysed in this chapter. Conclusions on the way these technologies can help achieve the desired results and their limitations are also drawn.

### 2.1 Motion capture technologies

The task of capturing human motion has been tackled in different ways. Technologies used span from image and video analysis to depth information structuring. The former shows successful results in controlled conditions, with several calibrated cameras. Image processing methods use

algorithms for, among others, flow detection and vector-machine decomposition. However, the raw Red-Green-Blue (RGB) data retrieved from cameras are prone to noise. To overcome this, techniques like classic image noise preprocessing, frequency filtering and background subtraction are used [29, 50].

Wearable optical markers in conjunction with a MoCap system (like *Vicon* [120]) gives more accurate results. These methods provide real-time, accurate motion data. The analysis software included in commercial systems ensures a complete, off-the-shelf solution to work with. However, these systems are expensive and not widely available. Furthermore, users must have markers attached at specific locations of their body (and often the assistance of the researcher is required) before using the application. These are major drawbacks if be used with elderly people, in nurseries or in private dwellings.

A potential solution is found in the gaming industry, where depth measurement and Time-of-Flight (ToF) devices bring interactive Human-Computer (H-C) communication to a different level. These devices are based on the principle of the time taken by a light particle (whose speed is constant and known) to travel from the source to the destination and back. The Microsoft Kinect device [105] uses this principle to produce a *depth map*, which is later used to segment the body in 31 parts, leading to a human body joints location estimation. A similar device, the Primesense Sensor [91], with the underlying OpenNI technology, provides a more extended specification. Outside light disturbance-insensitive precision is achieved by using an infra-red emitter and a Complementary Metal-Oxide Semiconductor (CMOS) sensor. The device is also supplied with an RGB camera and a microphone.

## 2.2 Motion alignment methods

Motion alignment is a growing research field in pattern recognition and behavioural analysis. Traditional approaches use motion templates to segment a motion data sample into different actions or behaviours [11, 82]. Body shape and colour cues are used in [102], along with cyclic motion analysis, to segment and classify human actions.

Cluster analysis is used in several motion segmentation solutions. Specifically, [68] uses k-means clustering and takes into account the sequentiality (in time) of the data, finding an optimal partition. A different approach is followed in [53], introducing a novel spatio-temporal Dimensionality

Reduction (DR) version of Isomap that takes into account the temporal inter-dependability of each data sample, unlike other DR methods based on Principal Component Analysis (PCA).

### 2.2.1 DTW

Dynamic Time Warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. While its applications in speech recognition can be found in works dated as early as 1975, Berndt and Clifford [8] first introduced the algorithm, based on dynamic programming, to find patterns in time series. The basic objective is to find an optimal frame-to-frame assignment between two *discrete* time series (of possible different length). The function to optimise (minimise) is the distance between the two alignments thus defined. This distance is often a compromise between the time displacement and the difference in absolute value between the two aligned data points.

A classic implementation of DTW can be seen in Algorithm 1. This implementation retrieves the minimum cost of the optimal alignment. The cost of all possible sub-alignments are computed in a look-up table (*DTW*), where  $DTW[i, j]$  stores the cost of aligning the  $i$ -th element of the series  $s$  with the  $j$ -th element of the series  $t$ . This cost is recursively computed with the previously calculated costs of all adjacent sub-paths.

---

#### Algorithm 1 Classical implementation of DTW

---

```

function DTWDistance(s: array [1..n], t: array [1..m])
   $DTW \leftarrow array[0..n, 0..m]$  ▷ Initialise cost matrix to maximum cost
  for  $i \leftarrow 1 \dots n$  do
     $DTW[i, 0] \leftarrow \infty$ 
  end for
  for  $i \leftarrow 1 \dots m$  do
     $DTW[0, i] \leftarrow \infty$ 
  end for
   $DTW[0, 0] \leftarrow 0$ 
  for  $i \leftarrow 1 \dots n$  do
    for  $j \leftarrow 1 \dots m$  do
       $cost \leftarrow d(s[i], t[j])$  ▷ Calculate cost of aligning  $s$ 's  $i$ -th frame with  $t$ 's  $j$ -th frame
       $DTW[i, j] \leftarrow cost + \text{minimum}(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1])$ 
▷ Insertion, deletion and matching costs respectively
    end for
  end for
  return  $DTW[n, m]$  ▷ Return the cost of the warping path
end function

```

---

Although the above implementation retrieves the cost of the optimal alignment (*warping path*), the latter can be reconstructed by backtracking, using the *DTW* table. Alternative implementations of DTW include restricting the look-up area to a determined window width. FastDTW [99] is an approximation of DTW that has a linear time and space complexity.

### 2.2.2 HACA

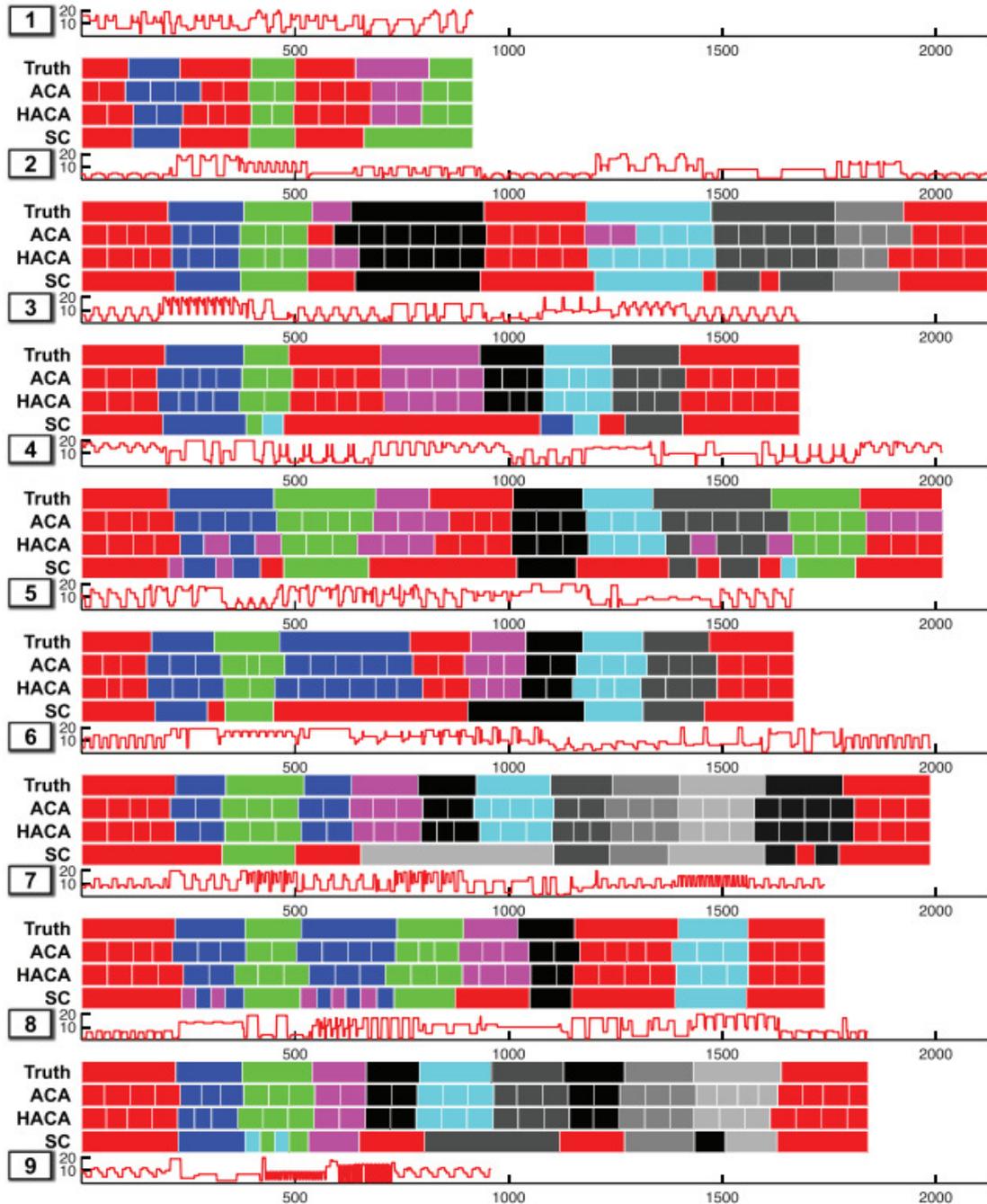
An unsupervised method for Hierarchical Aligned Cluster Analysis (HACA) is proposed in [127]. This uses  $k$ -means cluster analysis and a generalised DTW implementation to measure the equivalence between motion sub-segments. HACA finds a partition of a given multi-dimensional time series into  $m$  disjoint segments, such that each segment belongs to one of  $k$  clusters. It combines kernel  $k$ -means with the generalized dynamic time alignment kernel to cluster time series. Moreover, it provides a natural framework to find a low-dimensional embedding for time series. HACA is efficiently optimized with a coordinate descent strategy and dynamic programming. The HACA code (in MatLab) is available online [36].

One of the applications of HACA is the unsupervised segmentation of motion data samples. In order to test the robustness of the method, HACA is tested against some of the samples in the Carnegie Mellon University (CMU) Motion Capture database [21] to discover motion primitives. The human motion data are captured with a Vicon optical motion capture system (41 markers) and the motion data include absolute root position and orientation and the relative joint angles of 29 joints. Figure 2.1 illustrates the performance of HACA with real motion data samples from the aforementioned database.

## 2.3 Methods for approximate and exact string matching

In bioinformatics, sequence alignment forms part of the routinary search for new gene chain patterns [81]. Structural analysis of sequences of nucleotides or amino-acids is performed to optimise alignment between two or more of them. Pairwise alignment of sequences is done via dynamic programming implementation of algorithms like Needleman-Wunsch [83] or Smith-Waterman [107]. As an example application, the BLAST database [2] lets researchers perform an on-line search for similar DNA and RNA sequences, making use of some of these algorithms.

FIGURE 2.1: Example of segmentation of HACA with motion data samples from the CMU MoCap database. Motions are performed by subject 86. In the first row of each block, the frames are expressed as one of 20 labels given by  $k$ -means. The next four rows illustrate the ground truth and results given by, among others, HACA. White lines indicate the boundaries of actions, while the different colors correspond to distinct actions. With permission of Zhou, Feng [127]



String pattern matching [60], a method used on many applications, and string repetition detection [27] are two of many approaches related to the stated problem. However, the existence of an unbounded number of periods and the fact that some of them may contain missing and strange items will limit the outcome of these methods.

The tandem LCS (*Longest Common Subsequence*) problem [6, 116] consists of the identification of consecutive subsequences on a string matching a given pattern string. LCS [7] implicitly aims at inferring the subsequence with a higher degree of similarity with the pattern, given the number of needed edits to match the later with each subsequence of the sample string.

The cyclic string-to-string matching problem [17, 62, 73] differs from the analysis of periodic or repetitive sequences in the formulation of the problem. In this case, the sequences may be shifted so that the match is more obvious and it would lead to an equally acceptable solution. The application of this formulation can be found in object shape recognition.

Techniques for temporal analysis of events aims to extract analytical data from sequences of events in time in order to detect unusual behaviour of the involved agents. Intrusion detection in computer systems include approaches based on pattern matching [61], state transition analysis [52] and neural networks [32].

## **2.4 Assessment and adjustment of fitness performances given a reference**

Research in automatic assessment of physical performances is limited. The issue of providing of specific feedback to the user, in the context of basic fitness primitives, is not addressed.

In the field of tele-medicine, retrieved sensed data on the user's end are remotely sent to the expert physician or therapist. [63] is a holistic system that incorporates a network architecture to observe the evolution of patients performing function assessment exercises from their homes. The data shown to the specialist consist of a stream of higher level motion features like joint angles and joint range of motion (RoM) [85] and are sent alongside with a three-dimensional (3D) avatar reconstruction of the user's skeleton.

Ontologies are used to represent knowledge about both context and expert evaluation information. Some ontology-based approaches, like [9, 59, 123], rely on a set of rules drawn from expert advice and experimental conclusions that are triggered on request of the system, in search of a tailored feedback message. The latter often consists of an encouraging comment based on information like bio-feedback (e.g. calory intake or physical activity), user's personal data or user's level of activity. On the other hand, the approach described in [109] uses vibrating motors to provide feedback.

Motivation and level of engagement in health environments is considered in different fashions. Adaptation to the current level of performance is used in order to encourage the user and transmit a higher level of naturalness to the application. For example, the level of difficulty in exer-games can be adapted to the level of performance of the user [19, 89] or the avatar (representing oneself) can adapt its appearance [39, 58, 72]. Also, the nature of the exercises can change to adapt to the physical condition of the user [18]. Many systems exploit the capabilities and ease of use of mobile [59, 79] and wearable [109] devices.

Adjustment of natural human motion is performed in order to achieve various tasks such as manipulating robots [30, 33] or re-targeting [47] (i.e. re-scale a skeleton subject to certain constraints).

A human-like, synthetic avatar is often used as a guide for performance [10, 39, 58]. The demonstration is developed with the help of 3D animation engines and a list of predefined movements, sometimes with the help of motion capture methods such as optical markers or virtual reality gloves/suits [66].

Other reviewed approaches transform motion capture using Inverse Kinematics (IK) [67, 76], dynamics [76] or processing techniques [16].

## **2.5 Conclusions**

A series of methods for semi-automated motion capture, analysis and assessment have been summarised in this chapter. In order to put the current research into context, the task of performing automatic feedback production against exercise performance can be clearly subdivided into these three areas.

To the best of the authors' knowledge, none of the reviewed techniques address the stated problem as a whole. While the current research relies on a solidly developed motion capture framework (the Kinect camera), methods for motion alignment (needed in order to establish any kind of comparison between two or more performances), assessment and adaptation lack of adequate versatility, scalability and reachability in the addressed scenario (elderly people performing on their own).

Firstly, behaviour analysis and classification is ruled out as a way of analysing the motion prior assessment analysis. On one hand, a seemingly safe assumption is that the motion class to

which the exercise to be analysed belongs to is known beforehand. That is, one is expecting the individual to *repeat* or *mimic* a given movement or series of movements. Therefore, the key point would be to focus on reliably *matching* the common points of the two.

Secondly, human motion matching and alignment technologies assume a strong level of similarity between human motion samples of the same class. This is important, as a performance that is not *identical* may still be *similar* to the reference and, therefore, may still be matched in order to perform assessment. Likewise, current motion segmentation techniques do not deal well with the significantly smaller dimensionality of the data object of analysis and, again, are normally designed to process previously unknown motion data composed of complex sub-motions.

Finally, the area of research dealing with exercise assessment and feedback needs to be further explored. The results at the end of the process often consist of statistical data to be analysed by an expert, kinematic entropy and bio-feedback analysis. Moreover, the associated human-machine interfaces present some limitations in the level of expressibility that need to be tackled in order to deliver information to the targeted sector of population.

In conclusion, a new approach to tackle the reviewed limitations is needed to be developed and explored. To further review the research associated with the stated objectives, a survey on *ad-hoc* systems addressing a range of scenarios of human motion analysis and assessment is presented in Chapter 3.

## Chapter 3

# Technical review of ad-hoc approaches to human motion assessment

In this chapter, a series of works aiming at research on the enhancement of the quality of life of aged people is reviewed. Given the nature of this thesis, a review of Human-Computer (H-C) interfaces and three-dimensional (3D) human body tracking is also presented. The reviewed approaches are compared in terms of the different features used. Finally, the need for a new approach for automatic performance assessment is discussed.

### 3.1 Related applications

In Table 3.1, a series of reports found in the literature related to human motion capture is summarised and analysed in terms of their *context*, Motion Capture (MoCap) or *input method*, whether they are *markerless* or not, the kind of *body motion information* that they deliver, whether they are *real-time (RT)* systems or not, the kind of *feedback* delivered to (or gathered from) the user, the approach used for *motion recognition*, whether they keep a *history record* and whether they are *adaptable* or not. The reviewed systems target fields like *body pose estimation*, *human activity recognition and classification*, *dance education and rehearsal* and *physical activity performance*.

TABLE 3.1: Summary of reviewed motion capture and tracking approaches, with **M**=Markerless, **BMI**=Body motion information, **Feedback**=Feedback from/to the user, **MRM**=Motion recognition method, **HR**=History record and **A**=Adaptability

Ref.	Context	Input	M	BMI	RT	Feedback	MRM	HR	A
[105]	3D body joints positions calculation	Depth image	Yes	3D body joints	Yes	None	Body segmentation in parts. Decision forest classification	No	No
[101]	Human body pose tracking	Time-of-Flight (ToF), Kinect	Yes	16 body Points Of Interest (POIs)	Yes	None	Geodesic distances graph calculation. Target occlusion by optical flow calculation	No	No
[126]	Body pose estimation and labelling	Motion database, depth map	Yes	Surface mesh and body point cloud	No	None	Body pose model mapping. Skeletal information refinement	No	No
[44]	Human activity classification	Images	Yes	“Stickman” representation	No	None	Boundary-centroid calculation. Cyclical patterns recognition	No	No

Continued on next page

Table 3.1 Summary of reviewed motion capture and tracking approaches– Continued

Ref.	Context	Input	M	BMI	RT	Feedback	MRM	HR	A
[96]	Behaviour classification	Image	Yes	Optical flow	No	None	Key pose detection through flow changes analysis	No	N/A
[48]	Human activity surveillance	Mono. video	Yes	Blob silhouettes	Yes	None	Appearance models building	No	N/A
[113]	Repetitive movements discovery in human motion	3D optical markers	No	35 joints and ends	No	None	Normalisation. Euclidean distance	No	N/A
[71]	General purpose action recognition. Parkinson patients	MoCap datasets	N/A	Decomposition in primitive motion sequences	No	None	Decomposition based on activation	No	N/A

Continued on next page

Table 3.1 Summary of reviewed motion capture and tracking approaches– Continued

Ref.	Context	Input	M	BMI	RT	Feedback	MRM	HR	A
[3]	Human body motion classification	Video	Yes	two-dimensional (2D) body joints	Yes	None	Mass centre trajectory. Target occlusion by movement prediction	No	No
[104]	Traditional Japanese dance performance	Vicon	No	33 joints	No	None	Decomposition based on the speed of the hands and beat of the music	No	N/A
[112]	Real-time dancing companion	3D optical markers	No	3D 20 joints + 5 ends	Yes	Questionnaires	Progressive block matching with dance motion templates	No	Training and freestyle
[23]	Tai-Chi motion synthesiser	Exercise manuals, video	Yes	Key postures	No	None	Motion Index Table yielding motion annotation and further synthesis	No	N/A

Continued on next page

Table 3.1 Summary of reviewed motion capture and tracking approaches– Continued

Ref.	Context	Input	M	BMI	RT	Feedback	MRM	HR	A
[1]	Assessment of exercises in a tread mill	Video	Yes	Spine angle	Yes	None	Chamfer distance transform and temporal particle swarm-based contour search	No	No
[51]	Home exercise	Belt, static bicycle	No	Heart rate, speed and direction	Yes	Encouragement. Questionnaires	None	No	No
[97]	Reactive Virtual Trainer (RVT)	Vicon, ParleVision	No	3D hands and feet coordinates	Yes	Motivational messages	“Permitted trajectory” tracking of limbs centre of mass	Yes	Beat adjustment
[65]	Body tracking and clinical analysis on handset devices.	Mono. video	Yes	Body limbs joint angles	No	None	Body limbs segmentation based on a priori acknowledgement.	No	N/A

Continued on next page

Table 3.1 Summary of reviewed motion capture and tracking approaches– Continued

<b>Ref.</b>	<b>Context</b>	<b>Input</b>	<b>M</b>	<b>BMI</b>	<b>RT</b>	<b>Feedback</b>	<b>MRM</b>	<b>HR</b>	<b>A</b>
[86]	Elderly coaching of basic stretch exercises	Kinect	Yes	Feedback bar and alerts	Yes	Feedback bar and alert messages	Body limbs segmentation	Yes	No

[101] takes advantage of both ToF cameras and the Kinect device to detect regions of interest within a depth human body map and build up a skeletal structure with no other prior information. [105] takes a slightly different approach by seeking and assembling the human limbs independently, using solely depth information from the sensors and a vast human poses depth information database as ground truth. A similar approach is followed in [126], where the depth map is matched against a point cloud pose database.

In the field of human activity recognition, video sequence-based approaches are utilised in [44] and [96]. Activity classification is done through Singular Value Decomposition (SVD) of the flow followed by component analysis (like *key pose* identification). [48] describes an approach using surveillance cameras for human individuals tracking and movement monitoring by performing single monochrome or infrared (IR) video camera imagery processing. Appearance models of individuals are built and then used for tracking even in occlusive or cluttered scenarios. The work in [113] focusses on the identification of cyclic patterns in optical marker-based motion sequences. This is done by calculating a matching cost map between similar actions and discovering cyclic patterns within it. [71] analyses motion dataset samples to decompose human actions into shift-invariant basis functions. This allows a versatile, general representation of actions, thus reducing significantly the search scope and the complexity of the problem stated by the activity recognition procedure. Finally, an approach for human body joints motion tracking using video analysis can be found in [3], where a skeletal model is used to represent the movement to overcome the non-rigidity of the body structure. The tactic to reach a certain degree of both accuracy and efficiency is to measure the deviation of the feature points with respect to the initial frame. In order to analyse the behaviour of the participants, the location of the involved body limbs (hands and feet) are normalised with respect to their vertical coordinate components and statistically analysed through time.

Several educational dancing related applications have also been developed using MoCap technologies to assess users' performances. In [104], dance synthesis is applied in traditional Japanese dances. Data is acquired through optical markers and the Vicon system. Basic motions are extracted using the beat of the music and performers are assessed by their speed and centre of mass of their hands. [112] introduces a real-time dance companion and its application into ubiquitous dance teaching environment. Also using optical markers, the dances are first recorded in a database and the performance, instructed by an on-screen humanoid, is measured in terms of joint-wise cosine distance. This work also uses a continuous block matching cost-based algorithm to make the virtual trainer follow the user in real-time. This correspondence between two

different motions provides a suitable framework for exercise assessment. This system provides two levels of complexity: training and freestyle, thus incorporating adaptability to the user's capabilities. The same MoCap technology is used to develop a complex Tai-Chi synthesiser, which is described in [23].

Finally, in the field of physical activity assessment, an experiment involving the application of computer vision in physical activities performed by older people is discussed in [1], with the objective of assessing them on their performance from the point of view of the body posture. Specifically, video frames of exercise on a tread mill were recorded. Using statistical background subtraction, Chamfer distance transform and temporal particle swarm-based contour search, the inclination of the back (from a side view) and the shoulders (from a frontal view) were represented and recorded in an activity log. Users can follow their own progress and check if they were successful in keeping the right posture. [51] shows how technology can help athletes improve their training by incorporating a virtual agent in a stationary bicycle scenario. Using the heart rate as a feedback variable, levels of motivation and presence are evaluated using Intrinsic Motivation Inventory and ITC-Sense Of Presence Inventory (ITC-SOPI) methodologies. Results are interpreted for different circumstances (with/without virtual coach) and with different levels of immersion. The most reliable results can be achieved if MoCap hardware is incorporated, as shown in [97]. This work introduces a RVT in the context of supervised physical activities in a day-to-day environments. Scenarios like a normal rehearsal, post-internship rehabilitation, etc. are included. The envisioned virtual agent is taught how to show the exercises by previously recording a set of Trainer-Trainee interactions through optical markers motion capture. The RVT is also able to adjust the speed of the movements to the tempo of the audio input of the exercises.

### **3.2 Analysis of approaches and identification of improvements**

The above approaches encompass intrusive [51, 97, 104, 112, 113] approaches for gathering data, off-line [23, 44, 65, 71, 96, 104, 113, 126] reactive systems and non-adaptive [1, 3, 23, 44, 48, 51, 65, 71, 96, 101, 104, 105, 113, 126] methodologies. Moreover, few of the approaches deliver appropriate, direct feedback to the user and keeps a history record of the performance [86, 97].

None of the systems support all the desirable features for the case scenario of this research. For example, few systems are ready to work in real-time. Some were incomplete or have not been fully tested using realistic scenarios. Furthermore, not many of these systems take advantage of

the fact that the user can provide further information. That is, there is little feedback from the users that could otherwise be used as a more enriched source for information of their level of engagement, motivation and physical condition. Progression assessment is also a very desirable feature, since elderly people are often supervised by e.g. carers and relatives, who appreciate being able to track the improvement or worsening of their physical condition.

### **3.3 Conclusions**

None of the current approaches in Section 3.1 for human motion assessment are able to accurately compare two human motions, performed by different individuals. It is the belief of the author that an automatic system for simple physical exercise performance evaluation of a human, whose behaviour cannot be modelled or predicted *a priori*, is needed to achieve the above requisites.

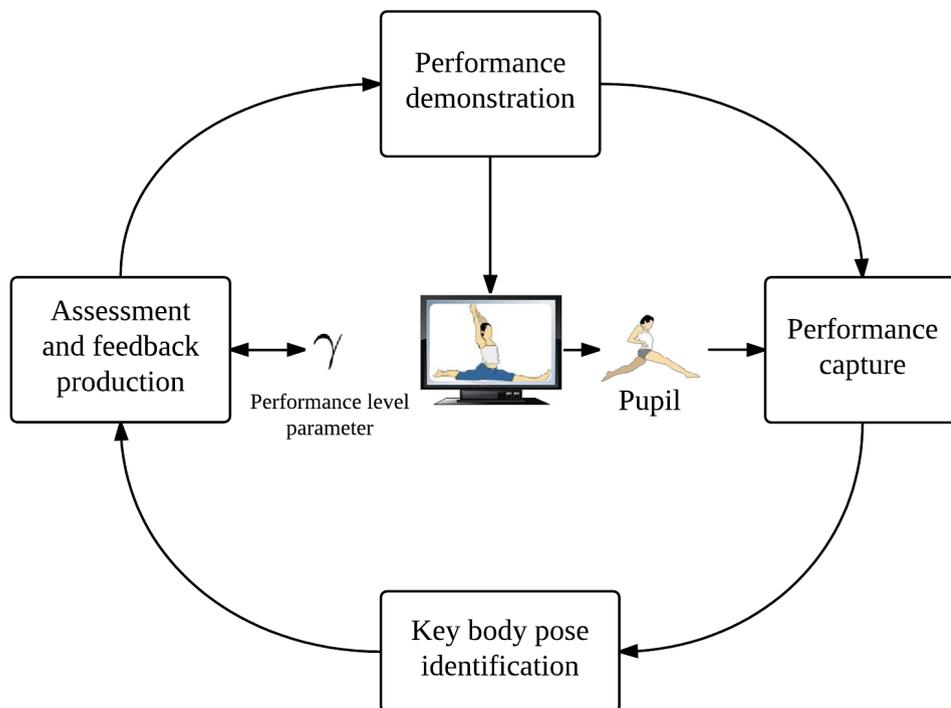
Moreover, body motion alignment techniques summarised in Section 2.2 present low performance on short, basic motions, like the ones intended for this research. These are liable for misclassification due to the repetitive and symmetric nature of the performances. For this reason, an approach that is not based on the whole motion data spectrum but on individual time momenta is needed. This issue is addressed and tackled in the proposed approach.

## Chapter 4

# System architecture

Figure 4.1 depicts the different components and processes of the system. Ideally, the user will engage in physical activities on a daily or weekly basis (depending on their condition) for a certain period of time. Analysis of each performance involves several stages.

FIGURE 4.1: Diagram representation of the system



The process flow begins with the user, who is introduced with an animation demonstration of the exercise. Then, their performance is captured, recorded and pre-processed. The first motion data analysis sub-task consists of identifying the *un-assessed landmarks*, that is, the *landmark*

*identification*. Finally, periodic landmark sequence analysis, motion synchronisation, landmark assessment and motion adjustment between the tutor's and user's performances are performed in the *assessment and feedback production* stage, in order to provide of assessment to the user. The performance level parameter ( $\gamma$ ) is used to represent the current level of the user and adjust both visual feedback and demonstration animation to their condition. These subsystems are described in more detail further in this thesis.

First, the features of the key body poses of the selected exercises are learnt from the training samples (a set of motion data samples extracted from a number of performances by users). This training data is stored for further use by the landmark identification subsystem, which attempts to infer key body poses from previously unseen motion data (i.e. the performances of the user). Once the landmarks are identified, both the motions of the user and the tutor can be matched for landmark assessment. Finally, upon the performance history of the individual, the system automatically adjusts the level of difficulty demanded by the exercises demonstration to suit the user so that the exercises are not too demanding to avoid discouragement.

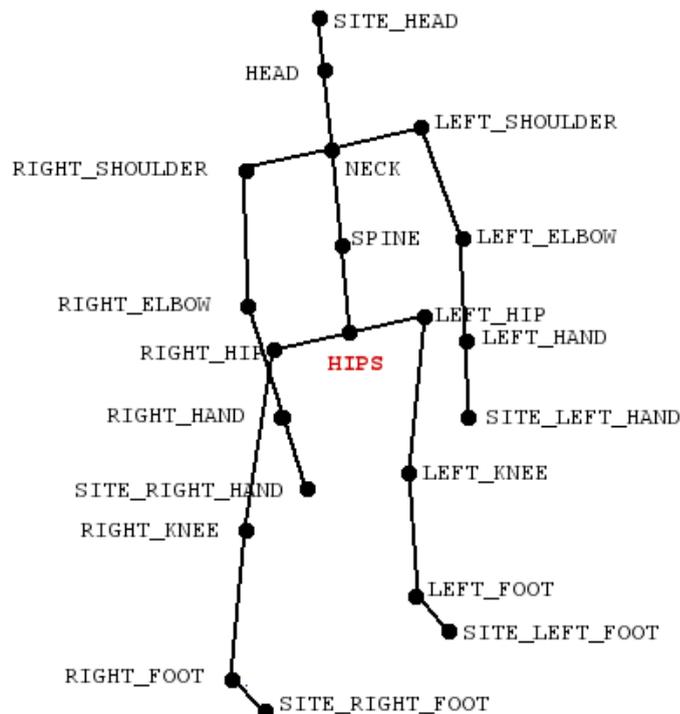
## 4.1 Hardware configuration

The motion data of both the tutor and the user are gathered using a Microsoft Kinect sensor. It is a device designed for the Microsoft Xbox gaming platform that has been used in numerous experiments in fields like gesture recognition [41, 95, 125], robotics [110], medical applications [22, 45] or education [117]. Kinect integrates a Red-Green-Blue (RGB) camera and a Time-of-Flight (ToF) infrared (IR) emitter and receptor. With the combination of the two, a depth image is built, in which every pixel contains a representation to the distance of its matched voxel to the device. A relatively accurate human motion tracking –compared with other non-invasive tracking devices– is produced by segmenting the body into 48 body points. In order to access to Kinect's data stream, a third party application, the Brekel Kinect software [13], is used. The OpenNI platform is used as middleware [5] for motion sequences extraction into Biovision Hierarchy (BVH)-formatted files [75].

A BVH-formatted file consists of a hierarchical data structure of joints and bones and its motion in time, shown in Figure 4.2. The first part defines, for each joint, an `offset` parameter (which stands for the translation along the  $x$ ,  $y$  and  $z$  axes with respect to the *parent* body joint, i.e., the one right above in the hierarchy). Note that the joint `offset` parameter also determines the

length of the child limb or, in the case of the HIPS, the initial translation with respect to the absolute zero coordinates (which, in turn, determines the overall translation of the skeleton). The second part lists the number of frames, the length (in seconds) of every frame and a line for each frame with the translation and rotation of the HIPS joint and the rotation of each joint with respect to their *parent* joint. An excerpt from a BVH sample file can be seen in Listing 4.1. Note that, for clarity, the body joint naming convention differs from that of Figure 4.2.

FIGURE 4.2: BVH file format joints hierarchy, with the *root* body joint (HIPS) highlighted in red



LISTING 4.1: Excerpt from the BVH file for the *p00\_ankles\_normal01* performance of the *Ankles stretch* exercise

---

```

HIERARCHY
ROOT Hips
{
  OFFSET 27.076 -36.483 -95.357
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT LeftUpLeg
  {
    OFFSET 9.674 -0.000 -0.000
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftLeg
    {
      OFFSET 0.000 -41.316 -0.000
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT LeftFoot

```

```

}
MOTION
Frames: 845
Frame Time:      0.0333333
-38.7839 32.5388 -315.236 0.169559 -1.95596 19.6741 5.9207 -7.63618 0.789524 -1.78613
  4.02118 -0.286727 -5.62184e-08 -4.70665e-08 5.96772e-07 0.741394 -4.69168
  0.0606484 -3.22932 8.88046 0.384683 7.06242e-08 -5.02152e-08 2.299e-07 -1.03744e
-07 -3.97613e-07 -5.85784e-06 0 0 0 5.38211 -11.7853 -20.4544 -16.0601 -14.2771
  4.8688 1.9825e-16 7.20232e-07 -3.15423e-08 0 0 0 -6.21678 -7.22445 64.531 -2.83135
 -17.3772 1.02894 8.1738e-07 1.52619e-08 -1.51113e-07 0.755509 3.89184 -0.0512864
 -1.3332e-08 7.13009e-08 -1.03486e-06
-38.6352 32.2042 -314.755 0.1867 -2.5017 18.9438 5.77729 -6.18004 0.624038 -1.72794
  3.14797 -0.220879 -1.13813e-07 8.59384e-08 -5.03787e-07 0.923192 -5.03058 0.080964
 -3.88326 11.3487 0.583672 -3.38877e-08 1.66899e-07 1.124e-06 -5.89202e-08
 -1.39128e-07 3.99873e-05 0 0 0 5.88813 -11.7125 -19.0164 -16.7056 -12.9497 5.14439
  1.76137e-07 -2.74889e-08 -4.45674e-08 0 0 0 -5.60858 -6.71399 66.0416 -3.85659
 -17.425 0.630139 -2.73737e-08 3.79408e-07 1.22204e-07 0.740747 3.7675 -0.0486754
 -1.32252e-08 9.33623e-09 -1.61563e-06

```

---

## 4.2 Motion data specification

The motion data, stored in BVH formatted files, is first pre-processed in order to build a point-of-view independent representation of the movement. In the following, a model of the human motion is described, in order to understand the problem that is being addressed. The specification of motion sequences is first shown, where the basic concepts of basic human motion are condensed. Then, theoretical background on body joint motion is explained. Finally, the geometrical interpretation of the previously described motion features is shown.

### 4.2.1 Motion sequences

A motion sequence is a representation of a human motion in terms of a set of specific joints and their exact location in time. Let  $j_i \in J$  denote the  $i$ -th body joint of a skeletal representation, with  $i = 1 \dots N_J$ . BVH format, for practical purposes, arranges the joints into a hierarchy, in which the HIPS joint is the *root* ( $j_{root}$ ) and

$$\text{Parent} : J \longrightarrow J^n \quad (4.1)$$

for some  $n \in \mathbb{N}$  defines the parent-child relationship among joints, so that  $\text{Parent}(j_i) = \{j_k\}$  are the joints immediately below  $j_i$  in the hierarchy.

For simplicity and in order to handle deterministic data, let us relax the time domain into the natural realm. Let us, then, define  $\mathbb{T} \in \mathbb{N}^{\mathbb{N}_T}$  as the time frame series of a given performance, and then

$$\mathfrak{M} : \mathbb{T}, \mathbb{J} \longrightarrow \mathbb{R}^n, \quad (4.2)$$

for some  $n \in \mathfrak{N}$  is a function class so that  $M(t, j_i) \in \mathfrak{M}$  is a feature value of the  $i$ -th body joint in *time frame*  $t$ . Specifically, let

$$P(t, j_i) \in \mathfrak{M} = \{p_x, p_y, p_z\}, \quad (4.3)$$

where, given a time frame  $t$  and a body joint  $j_i$ ,  $p_x$ ,  $p_y$  and  $p_z$  are the values of the  $x$ ,  $y$  and  $z$  *absolute positions*, respectively, of  $j_i$  in  $t$  within the Cartesian coordinates of the Motion Capture (MoCap)'s own world reference. Additional motion information of each joint in time is needed. That is,

$$O(t, j_i) \in \mathfrak{M} = \{o_x, o_y, o_z\}, \quad (4.4)$$

where  $o_x$ ,  $o_y$  and  $o_z$  are the *offset* values of the  $i$ -th body joint in the same  $x$ ,  $y$  and  $z$  Cartesian coordinates, respectively. The *offset* corresponds with the relative position with respect to  $\text{Parent}(j_i)$ . Likewise, let the translation information be defined as

$$Tr(t, j_i) \in \mathfrak{M} = \{tr_x, tr_y, tr_z\}, \quad (4.5)$$

where  $tr_x$ ,  $tr_y$  and  $tr_z$  are the translation values of the  $i$ -th body joint in  $x$ ,  $y$  and  $z$  Cartesian coordinates, respectively. Let

$$E(t, j_i) \in \mathfrak{M} = \{e_\phi, e_\theta, e_\psi\}, \quad (4.6)$$

where  $e_\phi$ ,  $e_\theta$  and  $e_\psi$  are the  $\phi$ ,  $\theta$  and  $\psi$  (or *heading*, *attitude* and *bank*) Euler angles values at time frame  $t$  for the  $i$ -th body joint. Finally, let

$$Q(t, j_i) \in \mathfrak{M} = \{q_w, q_x, q_y, q_z\}, \quad (4.7)$$

where  $q_w$ ,  $q_x$ ,  $q_y$  and  $q_z$  are the  $w$ ,  $x$ ,  $y$  and  $z$  unit quaternion values at time frame  $t$  for the  $i$ -th body joint.  $\text{Parent}(j_i)$  is used as the reference for the calculation of both  $E$  and  $Q$ .

## 4.2.2 Joint motion specification

From the theoretical basis of the motion sequence, it is important to define how these parameters are calculated for every joint. [43] describes a theoretical basis for algebraic and geometrical interpretation of joint-based robotic arms. An arbitrary rotation of a point in space  $p_{uvw} = (p_u, p_v, p_w)^T$  (with  $p_u$ ,  $p_v$  and  $p_w$  being the coordinates of  $p_{uvw}$ ) with respect to a fixed reference (the coordinate system  $OUVW$ ) is defined by a rotation matrix  $RM$ . The coordinates of the point after the rotation is defined as  $p_{xyz} = (p_x, p_y, p_z)^T$  and

$$p_{xyz} = RMp_{uvw}. \quad (4.8)$$

There are several different ways of expressing the rotation matrix  $RM$ . One of the most widespread methods is through Euler angles. Given the angular values  $\{\phi, \theta, \psi\}$ , a rotation sequence is defined as

- $\phi$  (or *heading*) about OW axis,
- $\theta$  (or *attitude*) about OU axis
- and  $\psi$  (or *bank*) about OV axis.

Each of these translations is independently defined as

$$R_\psi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{pmatrix}, \quad (4.9)$$

$$R_\theta = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (4.10)$$

and

$$R_\phi = \begin{pmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.11)$$

The composition of the three gives

$$RM = R_\psi R_\theta R_\phi. \quad (4.12)$$

Note that matrix product is non-commutative.

### 4.2.3 Geometrical interpretation of motion data

The information above is sufficient to perform a pairwise comparison of two motions, as discussed in the forthcoming sections. Nevertheless, in order to fully understand the meaning of the BVH files, it is worth to link it with the above explained theoretical foundation. First, the computation of the relative rotation matrix for each joint  $j_i$  and time frame  $t$  is straightforward. Given the specification in (4.9), (4.10) and (4.11), let us define the instantaneous rotation matrices as

$$\Delta R_\phi(t, j_i) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(E_\phi(t, j_i)) & \sin(E_\phi(t, j_i)) \\ 0 & -\sin(E_\phi(t, j_i)) & \cos(E_\phi(t, j_i)) \end{pmatrix}, \quad (4.13)$$

$$\Delta R_\theta(t, j_i) = \begin{pmatrix} \cos(E_\theta(t, j_i)) & 0 & -\sin(E_\theta(t, j_i)) \\ 0 & 1 & 0 \\ \sin(E_\theta(t, j_i)) & 0 & \cos(E_\theta(t, j_i)) \end{pmatrix}, \quad (4.14)$$

and

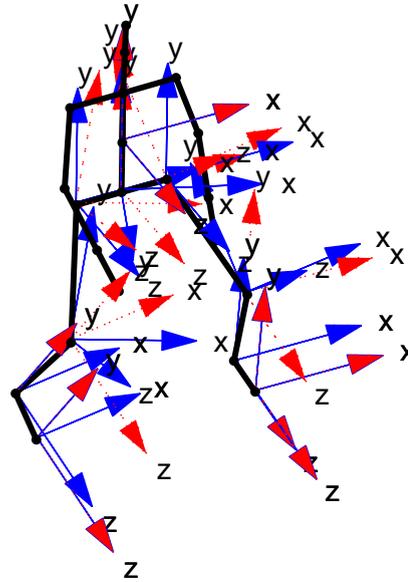
$$\Delta R_\psi(t, j_i) = \begin{pmatrix} \cos(E_\psi(t, j_i)) & \sin(E_\psi(t, j_i)) & 0 \\ -\sin(E_\psi(t, j_i)) & \cos(E_\psi(t, j_i)) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.15)$$

Thus,

$$\Delta RM(t, j_i) = \Delta R_\theta(t, j_i) \Delta R_\phi(t, j_i) \Delta R_\psi(t, j_i). \quad (4.16)$$

$\Delta RM(t, j_i)$  expresses the rotation angles of the  $i$ -th joint with respect to the three axes of an imaginary coordinate system, whose location and orientation at  $t$  depend in turn on  $\text{Parent}(j_i)$ 's location and orientation at  $t$ . Let us call this the *reference coordinate system*, with the origin in  $P(t, j_i)$  and the axes  $OUVW$  parallel to  $\text{Parent}(j_i)$ 's *local coordinate system*. An example of a skeleton and both reference and local coordinate systems of each body joint in the hierarchy is depicted in Figure 4.3.

FIGURE 4.3: Skeleton depicted with the reference (blue line) and local (red line) coordinate systems



In order to calculate the result of the rotation sequence for all  $j_i \in J$  in a determined time frame  $t$ , one can recursively define the absolute rotation matrix  $RM$  as the combination of the relative rotation of the joint (within the parent's absolute coordinate system) and the absolute rotation of the parent,

$$RM(t, j_i) = \Delta RM(t, j_i) RM(t, \text{Parent}(j_i)), \quad (4.17)$$

with

$$RM(t, \text{Parent}(j_{root})) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \forall t. \quad (4.18)$$

Furthermore, this rotation is applied after the joint has been translated with respect to its parent's coordinate system. Let this translation be

$$\delta P(t, j_i) = O(t, j_i) + Tr(t, j_i). \quad (4.19)$$

After the relative translation and the absolute rotation, the absolute position of the parent should be added to the transformation, leading to

$$P(t, j_i) = \delta P(t, j_i) RM(t, j_i) + P(t, \text{Parent}(j_i)). \quad (4.20)$$

The relative position of the joint within the reference coordinate system is computed as

$$\Delta P(t, j_i) = \delta P(t, j_i) \Delta RM(t, j_i). \quad (4.21)$$

Finally, in order to calculate the quaternion values  $w$ ,  $x$ ,  $y$  and  $z$ , given  $\phi = E_\phi(t, j_i)$ ,  $\theta = E_\theta(t, j_i)$  and  $\psi = E_\psi(t, j_i)$ ,

$$\begin{aligned}
Q_w(t, j_i) &= \frac{\sqrt{1 + \cos(\phi)\cos(\theta) + \cos(\phi)\cos(\psi) - \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\theta)\cos(\psi)}}{2} \\
Q_x(t, j_i) &= \frac{\cos(\theta)\sin(\psi) + \cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi)}{4Q_w(t, j_i)} \\
Q_y(t, j_i) &= \frac{\sin(\phi)\cos(\theta) + \sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi)}{4Q_w(t, j_i)} \\
Q_z(t, j_i) &= \frac{-\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\theta)}{4Q_w(t, j_i)}.
\end{aligned}
\tag{4.22}$$

### 4.3 Conclusions

The overall system architecture of the proposed approach has been presented in this chapter. A procedural design of the envisioned sub-stages, addressing the issues reviewed in the preceding sections, has been summarised and will be further explained in the forthcoming chapters. Finally, basic specifications on the hardware configuration, as well as the format and meaning of the motion data to be analysed, has been introduced.

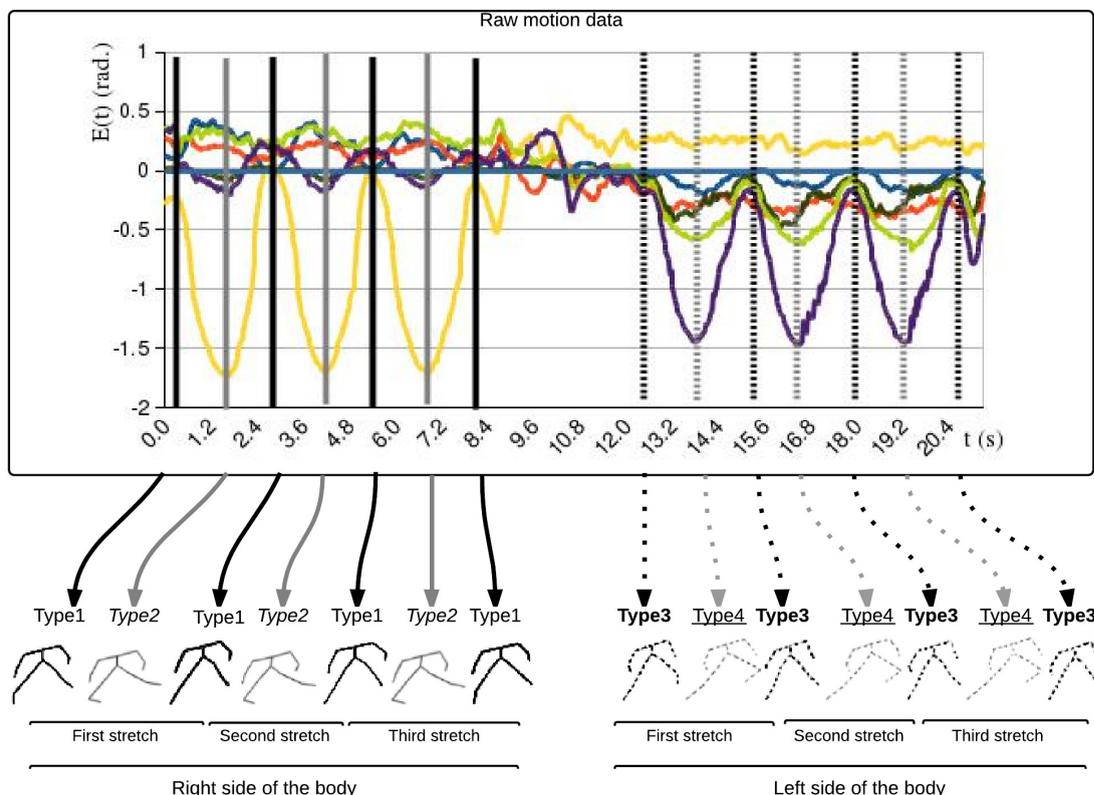
## Chapter 5

# A method for automatic key body pose identification in motion capture data of physical exercises

This chapter introduces a novel, multi-class machine learning approach for the identification of key human body poses. The system is divided into landmark model training and landmark identification stages. Firstly, a Biovision Hierarchy (BVH) file, encoding a motion sequence, is read and the motion data is extracted and smoothed. Secondly, Principal Component Analysis (PCA) is used to reduce the dimensionality of the input data, producing the motion signature, a low-dimensionality, multi-valued data series. This procedure is applied to both training and testing samples. The system first learns the features of the training set, prior transformation of the training motion signatures by frame-wise features extraction. Then, testing samples are processed in order to discover body poses (candidate landmarks) within the motion likely to correspond with the learnt key body poses (ground truth landmarks). Finally, the candidate landmarks set is refined using clustering methods.

Each landmark is assigned a landmark type, each of which, within each motion class, represents a milestone (i.e., a body pose of interest within each end of a range of motion (RoM) stretch). Figure 5.1 shows a representation of a motion sample from the *Ankles* exercise. The ground truth landmarks (i.e. manually annotated key body poses) are shown by means of vertical lines within the motion data and body postures representation.

FIGURE 5.1: Skeletal and plot representation of an instance of the *Ankles* exercise, its motion data and the ground truth landmarks as vertical markers. Landmark types are labelled above each skeletal pose as *TypeN*

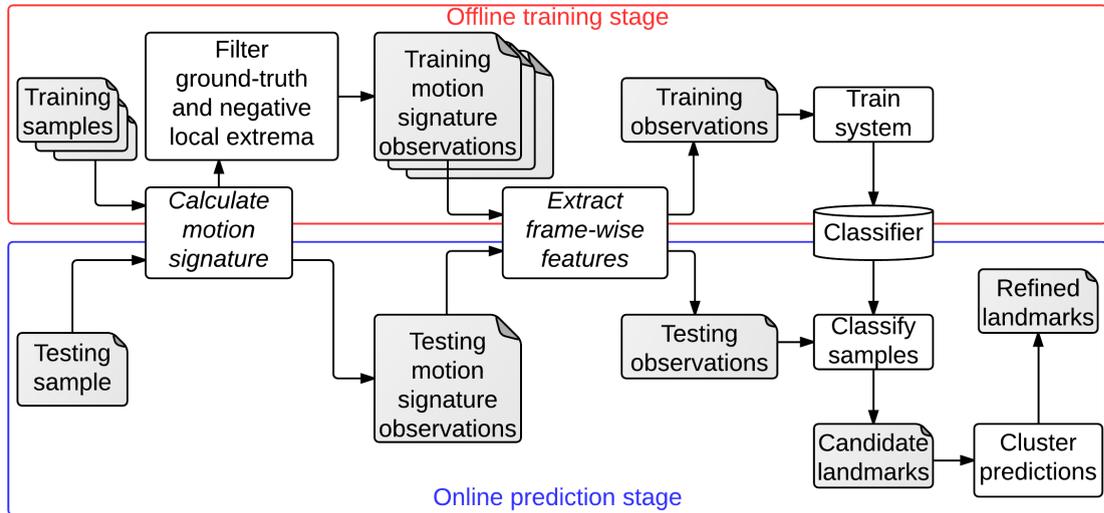


## 5.1 Overview of the method for landmark identification

The first two stages of the performance assessment system consist of learning the features of previously annotated motion data (training samples) and identifying key body poses from unseen samples (testing samples). A selected group of training samples from regular performers form the training set. The exercise performances are captured by the Motion Capture (MoCap) device and the data is stored in BVH formatted files (see Section 4.1). These samples are manually annotated, identifying, for each landmark, its landmark type and its assessment level. As part of the landmark identification and landmark assessment stages, the system learns the features of all landmarks of each training sample. The *key body pose learning and identification* subsystem (see Figure 5.2) makes use of these learnt features to classify new samples.

During the off-line training stage, the key body pose classifier is trained with the training observations, extracted from the training samples. Training samples are transformed into a series of instances or training motion signature observations, one per frame. In order to create a

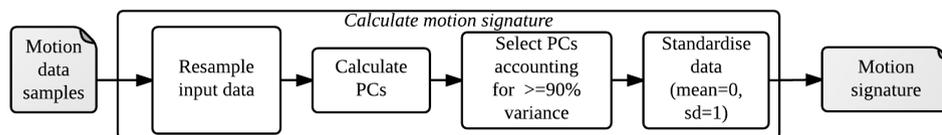
FIGURE 5.2: Key body pose learning and identification subsystem



good set of training observation for the classifier, the observations are filtered so that a balanced subset of both positive and negative observations are produced. That is, the system has to filter ground truth and negative local extrema from the motion signature observations. Then, the training observations are produced by transforming the filtered motion signature observations into frame-wise features.

In the case of the testing observations, the filtering stage is skipped, since every frame may contain a potential landmark. Otherwise, the testing motion signature observations and the testing observations are calculated using the same processes as the training observations. After the initial classification (candidate landmarks), the results are refined by use of clustering methods and the refined landmarks are finally produced.

FIGURE 5.3: Motion signature calculation diagram



Both stages make use of two common procedures, *Calculate motion signature* (shown in Figure 5.3) and *Extract frame-wise features*. The former transforms the input motion data (consisting of real value data series, representing absolute positions or relative angles of each involved body joint) into the dimensionality-reduced motion signature (a series of mutually uncorrelated data series, accounting for, at least, 90% of the variance of the motion data). This is done by first re-sampling the data and calculating, selecting and standardising its Principal Components (PCs). Standardisation of the PCs is achieved by transforming the data into an equivalent distribution

with zero mean and standard deviation of 1. The frame-wise features are a series of feature values per frame that depend on the values of each of the training motion signature observations.

## 5.2 Motion signature calculation

For every motion class, the *dimensionality*  $D$  is defined as the product of the *number of involved body joints* ( $N_J$ ) and the *number of involved Degrees Of Freedom (DOFs)* (i.e. coordinate axis per joint,  $N_{DOF}$ ), so that  $D = N_J * N_{DOF}$ . Exercises with a different value for  $D$  have been used. The motion data extracted straight from the MoCap device (BVH files) often contains noise, which leads to challenges, since the motion features later extracted depend on the curvature of the signal, more specifically, on the location of local extrema. Therefore, a more refined input by smoothing is required.

For a given exercise performance  $i$ , the extracted row motion data series  $E$ ,  $P$  and  $Q$  (see Section 4.2.1). A different *motion data source*  $M^P$  has been chosen for each different exercise, that can take the values of either  $E$  or  $P$ , as the basis for calculation of the landmark identification features. The reason for this decision is that that the approach gives better results if different motion data sources are selected for each motion class.

### 5.2.1 Data smoothing

As mentioned, the motion data often contains undesirable elements like noise or inconsistencies due to the limited accuracy of the capture technologies like self occlusion and low frame rate. To overcome these undesirable effects, each feature vector of a given motion data source  $M^P$  computed in the previous stage is re-sampled at a certain time window  $\omega$  and then recovered with a *cubic spline* in order to match the original sample length. Let

$$C(M^P) = \text{CubicSpline}(\text{ReSample}(M^P, \omega, |M^P|)) \quad (5.1)$$

be the smoothing function for any motion data series  $M$  and  $C = C(M^P)$  be the re-sampled result of the motion data.

### 5.2.2 Dimensionality reduction

PCA provides a means to reduce the dimensionality of a given set of data series. The idea behind it is to apply an orthogonal transformation to the original data (composed by possibly correlated variables) into a set of values of linearly uncorrelated variables (the PCs). The dimensionality of the chosen PCs may be equal or smaller than that of the original data. Let  $X^{n \times p}$  be a matrix representation of the data, with  $n$  samples and  $p$  variables. A series of  $p$ -dimensional vectors (*loadings*)  $W_k$  will map each row vector  $X_i$  to the PCs (or *scores*)  $T_i$ , so that

$$T = X W. \quad (5.2)$$

$W$  is often referred to as the *eigenspace* representation of  $X$ . One of the methods to calculate PCA's loading matrix is Singular Value Decomposition (SVD), a matrix factorisation that defines the decomposition of  $X$  as

$$X = U \Sigma W^T, \quad (5.3)$$

where  $\Sigma^{n \times p}$  is a diagonal matrix of positive numbers (*singular values* of  $X$ ) and each column of both  $U^{n \times n}$  (*left singular vectors* of  $X$ ) and  $W^{p \times p}$  (*right singular vectors* of  $X$ ) is an orthogonal unit vector. Then,

$$T = U \Sigma. \quad (5.4)$$

The proposed approach is to calculate the PCs' loadings of the tutor and use them to calculate the *eigenspace* representation of each performance. Each sample (user's performance) is represented by the smoothed motion data matrix  $C^{N_{\mathbb{T}} \times N_{\mathbb{J}}}$ , previously defined.

Let  $V$  be the matrix form of the loadings thus defined with the given samples of the tutor of an exercise. The motion signature of a given performance of the same exercise is then defined as a multi-variable time series in the associated *eigenspace*, as *the most representative eigenspace vector transformation* of  $C$ . The *motion eigenspace matrix*  $EIG$  is computed as

$$EIG = V_k \cdot (C - \vec{C}_k), \quad (5.5)$$

where  $\vec{C}_k$  is a vector with the feature means of the tutor.  $EIG$  is the eigenspace matrix representation of a motion sample. Each vector  $EIG(l)$  corresponds to a PC with expected

$$\rho_l = e_l / \sum_{j=1}^{3N_j} e_j \quad (5.6)$$

level of representation, provided that  $C$  belongs to the same exercise class and that the samples follow a normal distribution. For this project, an *accumulated level of representation* of 90% has been chosen, i.e.

$$sig = \{EIG(1), EIG(2), \dots, EIG(\Delta_k)\}, \quad (5.7)$$

where

$$\Delta_k = \arg \min_n \sum_{i=1}^n \rho_i \geq 0.9. \quad (5.8)$$

Figure 5.4 depicts the first stage of the system of smoothing the input data and transforming it into a lower-dimensionality time series (motion signature).

### 5.3 Selection of training observations and testing observations

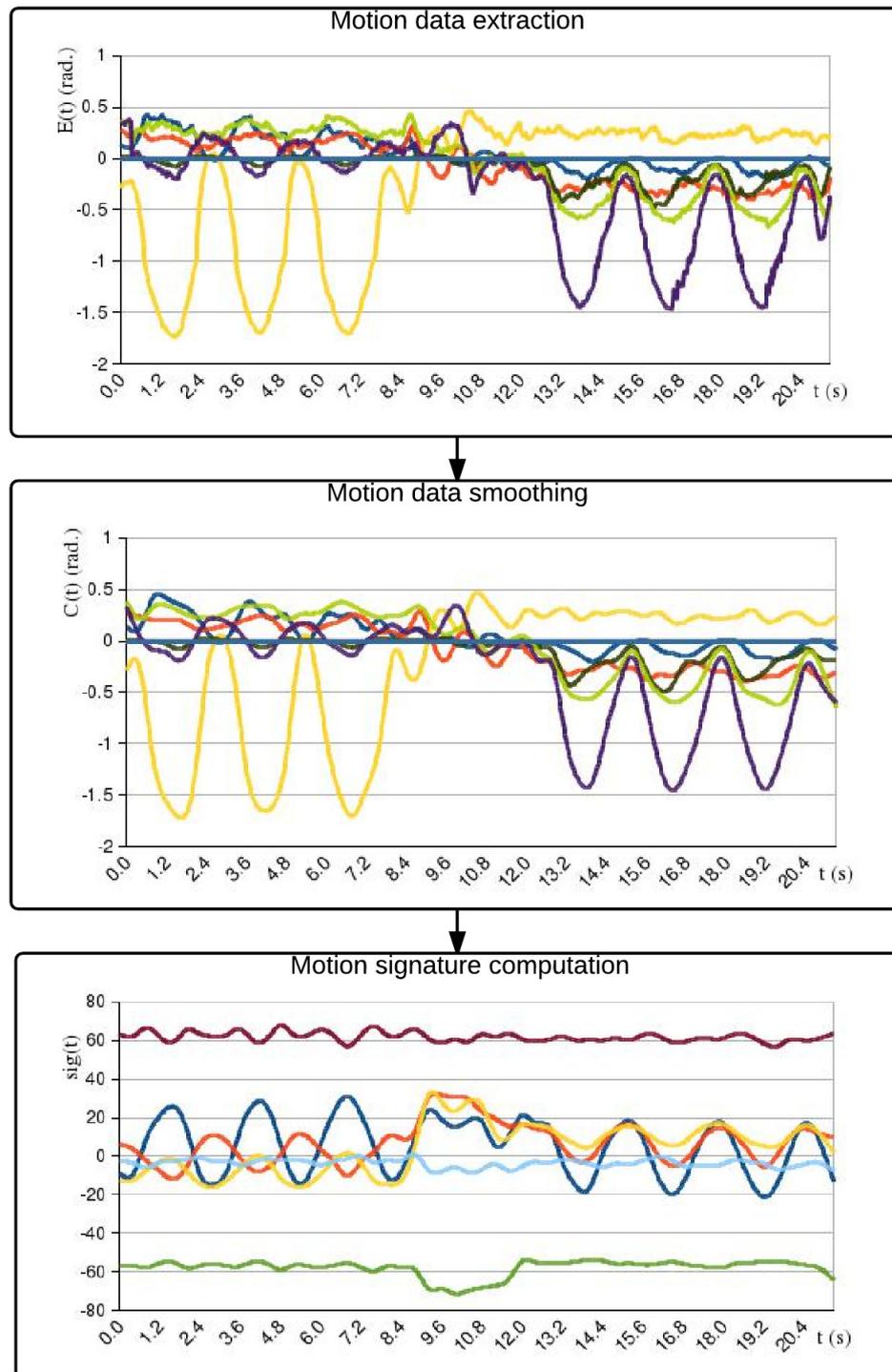
Let  $\mathcal{O} = \{x_1, x_2, \dots, x_n, \vec{y}\}$  be a single observation, made up of  $n$  features, for all  $x_i$ , with  $i = 1..n$ , and an observation class vector  $\vec{y} \in Y$ . A number of these observations are built from the motion data and form the basis for learning the model. As pointed out in Section 5.1, prior offline learning of the features of the training samples  $\mathcal{O}^{Tr}$  is required. Subsequently, upon learning the model, key body poses are identified from the testing observations  $\mathcal{O}^{Ts}$ , those extracted from the motion data of the sample to be evaluated.

Let  $\chi^G = \{\chi_1 \dots \chi_N\}$ , with

$$\chi_i = \{\text{FRAME}(\chi_i), \text{TYPE}(\chi_i), \{\text{AL}(\chi_i, j_i)\}\} \quad (5.9)$$

and

FIGURE 5.4: Pre-processing diagram. Raw motion data  $M(t)$ , smoothed motion data  $C(t)$ , motion signature  $sig(t)$  and ground truth landmarks of a sample of the *Ankles* motion class. Relative angle values (in *rad.*) of each joint and DOF (in the case of  $M$  and  $C$ ) and PCs (for  $sig$ ) are depicted with different line markers



$$\text{FRAME}(\chi_i) < \text{FRAME}(\chi_{i+1}) \forall \chi_i \in \chi^G, \quad (5.10)$$

be the ground truth landmarks of a given sample.

Let  $\mathbb{T}s = \{1 \dots N_{\mathbb{T}s}\}$  be the set of selected *testing frames* and

$$\mathbb{T}r = \{\text{FRAME}(\chi^G)\} \cup \{\text{Random}(\mathbb{T}s - \text{FRAME}(\chi^G), 3 \cdot |\chi^G|)\} \quad (5.11)$$

be the *training frames*, where  $\text{Random}(R, h)$  selects  $h$  distinct elements randomly, out of the  $R = \{r_1 \dots r_N\}$  set. Then, the training observations and testing observations will be the output of a function of  $\mathbb{T}r$  and  $\mathbb{T}s$ , respectively.

In other words, the testing observations consist of all of the sample frames of the testing sample, while the training observations only includes the ground truth frames corresponding to the ground truth landmarks and an equally-sized subset of other random frames of each training sample.

## 5.4 Candidate landmark identification

This stage consists of the identification of candidate landmarks ( $\chi$ ) that are more likely to represent a key body pose of the exercise. A classifier is built from the training set. Training observations correspond to a balanced set of positive (ground truth landmarks) and negative (local extrema that are *not* ground truth landmarks) observations of each training sample. The testing observations are produced from the testing samples. The former are fed to the trained classifier, which labels each testing observation with either a specific landmark type or *not a landmark candidate*.

### 5.4.1 Incompleteness and symmetry of landmark sequences

For a (possibly) incomplete sequence  $\bar{\chi}_P$  of identified landmarks and a reference sequence  $\chi^G_T$  of the tutor's ground truth landmarks, two important aspects should be considered. First, the number of identified landmarks may be incomplete, or the landmark sequence repetitiveness is different, i.e.

$$|\{\chi_i \in \chi^G_T / \text{TYPE}(\chi_i) = \tau_k\}| \geq |\{\chi_j \in \bar{\chi}_P / \text{TYPE}(\chi_j) = \tau_k\}| \forall \tau_k \in \text{TYPE}(\chi^G_T). \quad (5.12)$$

Second, the landmark sequence symmetry of the performance may also be different and, for some subsequence on the left hand side of  $\chi^G_T$ , the synchronised landmarks of  $\bar{\chi}_P$  may be at the end of the ordered sequence, or vice-versa. That is,

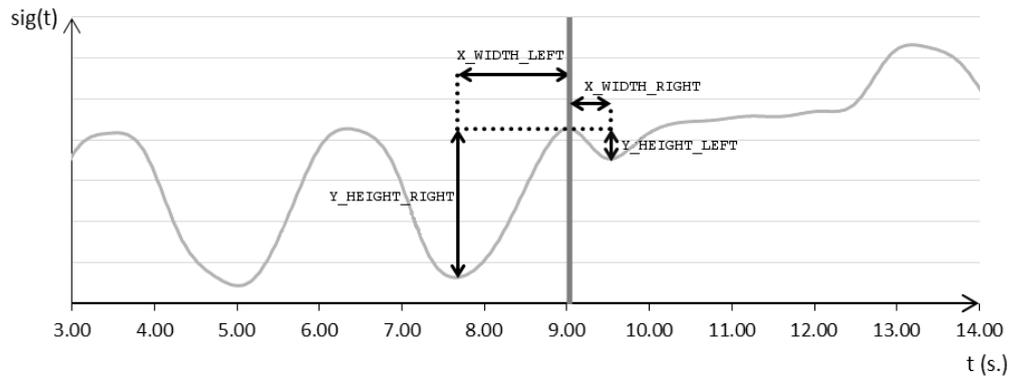
$$\exists \chi_i \in \chi^G_T / \max(\text{FRAME}(\{\chi_i \dots \chi_n\})) < \min(\text{FRAME}(\{\chi_1 \dots \chi_{i-1}\})). \quad (5.13)$$

This shows the need for a novel frame-wise key body pose identification system that does not stick to tight assumptions for the format of the motion data samples.

### 5.4.2 Frame-wise feature extraction

In order for a motion signature  $sig$  to be interpretable by the classifier, it has to be transformed into a series of observations, one per frame. For each observation, a set of features is defined, based on characteristics of the curve of the motion signature in the neighbourhood of the frame.

FIGURE 5.5: Example of extracted frame-wise features. The dark grey vertical marker indicates the frame whose features are being calculated



The frame-wise features consist of the distances to the closest signal extrema on both sides, as shown in Figure 5.5.

Let

$$Pk(s) = \{t \in \{1..|s|\} / s(t-1) < s(t) \leq s(t+1) \text{ or } s(t-1) \leq s(t) < s(t+1)\} \quad (5.14)$$

be the local extrema frames of a given data series  $s$  and let

$$PkL(s, t_i) = \max(t \in Pk(s) / t \leq t_i) \quad (5.15)$$

and

$$PkR(s, t_i) = \min(t \in Pk(s) / t \geq t_i) \quad (5.16)$$

be the closest local extrema of a given frame  $t_i$  to the left and right, respectively, within a series  $s$ . Then, the frame-wise feature functions X\_WIDTH\_LEFT, X\_WIDTH\_RIGHT, Y\_HEIGHT\_LEFT and Y\_HEIGHT\_RIGHT are defined as

$$\begin{aligned} X\_WIDTH\_LEFT(sig, t, j_i) &= \|PkL(sig(j_i)) - t\| \\ X\_WIDTH\_RIGHT(sig, t, j_i) &= \|PkR(sig(j_i)) - t\| \\ Y\_HEIGHT\_LEFT(sig, t, j_i) &= sig(t, j_i) - sig(PkL(sig(j_i)), j_i) \\ Y\_HEIGHT\_RIGHT(sig, t, j_i) &= sig(t, j_i) - sig(PkR(sig(j_i)), j_i), \end{aligned} \quad (5.17)$$

where  $\|\cdot\|$  is the absolute value. The transformation of a motion signature  $sig$ , given a series of frames  $\mathbb{T} = \{t_1 \dots t_{N_T}\}$ , into frame-wise features observation  $\mathcal{O}^{FWF}$  is then defined as

$$\begin{aligned} \mathcal{O}^{FWF}(sig, \mathbb{T}, \text{TYPE}_{\chi_G}) &= \{t, \\ &X\_WIDTH\_LEFT(sig, t, j_i), X\_WIDTH\_RIGHT(sig, t, j_i), \\ &Y\_HEIGHT\_LEFT(sig, t, j_i), Y\_HEIGHT\_RIGHT(sig, t, j_i), \\ &\{\text{TYPE}_{\chi_G}(t)\} \\ &\} \forall t \in \mathbb{T}, j_i \in \mathbb{J} \end{aligned} \quad (5.18)$$

for each frame  $t$  and body joint  $j_i$ . The  $\text{TYPE}_{\chi^G}$  function is defined as the landmark type descriptor, being

$$\text{TYPE}_{\chi^G}^{\mathbb{T}r}(t) = \begin{cases} \text{TYPE}(\chi^G_i), & \text{if } \exists \chi^G_i \in \chi^G / \text{FRAME}(\chi^G_i) = t \\ \text{none}, & \text{otherwise} \end{cases} \quad (5.19)$$

in the case of the training observations, and

$$\text{TYPE}_{\chi^G}^{\mathbb{T}s}(t) = \text{none} \quad (5.20)$$

in the case of the testing observations.

That is, given a series of motion signatures  $\vec{sig}^{\mathbb{T}r}$  of training samples  $sig_i^{\mathbb{T}r} \in \vec{sig}^{\mathbb{T}r}$ , their ground truth landmarks  $\vec{\chi}^G = \{\chi^G_i\}$  and their selected training frames  $\vec{\mathbb{T}r} = \{\mathbb{T}r_i\}$ , the training observations are chosen as

$$\mathcal{O}^{\mathbb{T}r} = \{\mathcal{O}^{FWF}(sig_i, \mathbb{T}r_i, \text{TYPE}_{\chi^G_i}^{\mathbb{T}r})\} \forall i. \quad (5.21)$$

Equivalently, given the motion signature  $sig^{\mathbb{T}s}$  of testing sample, its ground truth landmarks  $\chi^G_{\mathbb{T}s}$  and its selected testing frames  $\mathbb{T}s$ , the testing observations are chosen as

$$\mathcal{O}^{\mathbb{T}s} = \mathcal{O}^{FWF}(sig^{\mathbb{T}s}, \mathbb{T}s, \text{TYPE}_{\chi^G_{\mathbb{T}s}}^{\mathbb{T}s}). \quad (5.22)$$

Some frame-wise features, for certain frames located at both ends of the data series, cannot be calculated, since there is no local extrema on either side. In this situation, the relevant features are set as *unknown* to represent missing data.

### 5.4.3 Landmark identification

Let

$$\chi = \{\{\text{FRAME}(t), \text{TYPE}(t)\} \forall t \in \mathbb{T}s\} \quad (5.23)$$

be the candidate landmarks classified by the landmark identification sub-process as positive samples (i.e., landmark type different from *none*). Let

$$\Phi(\vec{x}^{\mathbb{T}r}, \vec{y}^{\mathbb{T}r}, \vec{x}^{\mathbb{T}s}) = y^{\mathbb{T}s} \quad (5.24)$$

with  $y^{\mathbb{T}s} \in Y$ , be the definition of a general purpose machine learning classifier  $\Phi$  as a function of the  $n$  feature values of  $N_{\mathbb{T}r}$  training observations  $x^{\mathbb{T}r}$ , their observation class label values  $\vec{y}^{\mathbb{T}r}$  and the  $n$  feature values of the testing observations  $x^{\mathbb{T}s}$ .

Let  $\mathbb{F}(\mathcal{O}^{\mathbb{T}r}, o) = y$ , with  $y \in Y$ , denote the output of a multi-class machine learning classifier trained with training observations  $\mathcal{O}^{\mathbb{T}r}$  over an unseen testing observation  $o$ , so that

$$\mathbb{F}(\mathcal{O}^{\mathbb{T}r}, o) = \Phi(\{\{o_j(i)\}\}, \{\{\text{TYPE}(o_j(i))\}\}\{o(i)\}) \forall o_j \in \mathcal{O}^{\mathbb{T}r}, i = 2..n. \quad (5.25)$$

Then,  $\chi$  is defined as

$$\chi(\mathcal{O}^{\mathbb{T}r}, \mathcal{O}^{\mathbb{T}s}) = \{\{\text{FRAME}(o) = o(1), \text{TYPE}(o) = \mathbb{F}(\mathcal{O}^{\mathbb{T}r}, o)\} \forall o \in \mathcal{O}^{\mathbb{T}s}\}. \quad (5.26)$$

#### 5.4.4 Machine learning classifiers

The Weka framework [40] has been used to build the learning system. Specifically, C4.5 [93], support vector machines (SVM) [24] and Naive Bayes [54] have been selected as independent machine learning classifiers for each exercise. In order to improve the performance, the AdaBoost [42] meta-machine learning classifier has also been considered.

- C4.5: a decision tree is built based on training data, by constructing production rules on each non-leaf node, each of which is concerned with the value of a specific attribute of a feature vector, and producing a class value at the root-level. In this work, Webb's grafted J48 [121] implementation in Weka is used. Default parameters are used: confidence threshold for pruning of 0.25, 2 instances per leaf (minimum), 3 folds for reduced error pruning (one of them as pruning set) ...
- Support vector machines: in binary classification problems, support vectors are calculated in order to split the data into two, by using hyperplanes that optimise the error rate of

the classifier. Nonlinear kernel transformations of the input data allows for nonlinear classification. Platt's [55] Sequential Minimal Optimisation (SMO) implementation in Weka (with a complexity  $-C-$  parameter value of 1.0 , no internal cross-validation and no normalisation,  $10^{-3}$  tolerance and  $\epsilon = 10^{-12}$ ), using a polynomial Kernel (with a cache size of 250007 and 1.0 exponent), is selected. Parameter values are default values of the framework. Multi-class problems are solved using pairwise classification (1-vs-1) and, if logistic models are built, pairwise coupling, according to [49].

- Naive Bayes: a probabilistic classifier inspired by the Bayes theorem with strong independence between features. Weka's NaiveBayes classifier, using estimator classes, is used.
- AdaBoost: a meta-algorithm: it works on the ground of a "weak classifier" and its aim is to improve its performance by setting up several classifiers, each of which focus on a different training sub-set (those that were not optimally classified by previous classifiers). An implementation of the AdaBoost.MV [20] modification of Weka's original AdaBoost [42], which uses validation sets to attempt preventing over-fitting, is used.

## 5.5 Landmark refining

The outcome of the previous stage often contains candidate landmarks that need to be discarded, namely several landmarks in the neighbourhood of each ground truth landmark, rather than just one. A subset  $\bar{\chi} = \bar{\chi}(\chi) \subset \chi$  is selected as the final candidate landmarks. The approach developed is to cluster observations of  $\chi$  that are strongly related and then choose the best candidate in each landmark cluster. The features analysed for each candidate landmark are the values of the motion signature in the corresponding frame, i.e.

$$\bar{\chi}(\chi) = \text{SelectFromClusters}(\text{Cluster}(\chi, sig), sig). \quad (5.27)$$

In other words,  $\bar{\chi}$  is the result of:

- first, clustering the candidate landmarks  $\chi$  into  $N_\sigma$  clusters, so that

$$\text{Cluster}(\chi, sig) = \{\sigma_r = \{\chi_j \in \chi\}\}, r = 1..N_\sigma, \quad (5.28)$$

and

- then, choosing one  $\chi_j$  from every  $\sigma_r$  as the final refined landmark

A depiction of both landmark identification and landmark refining stages is shown in Figure 5.6.

### 5.5.1 Landmark clustering methods

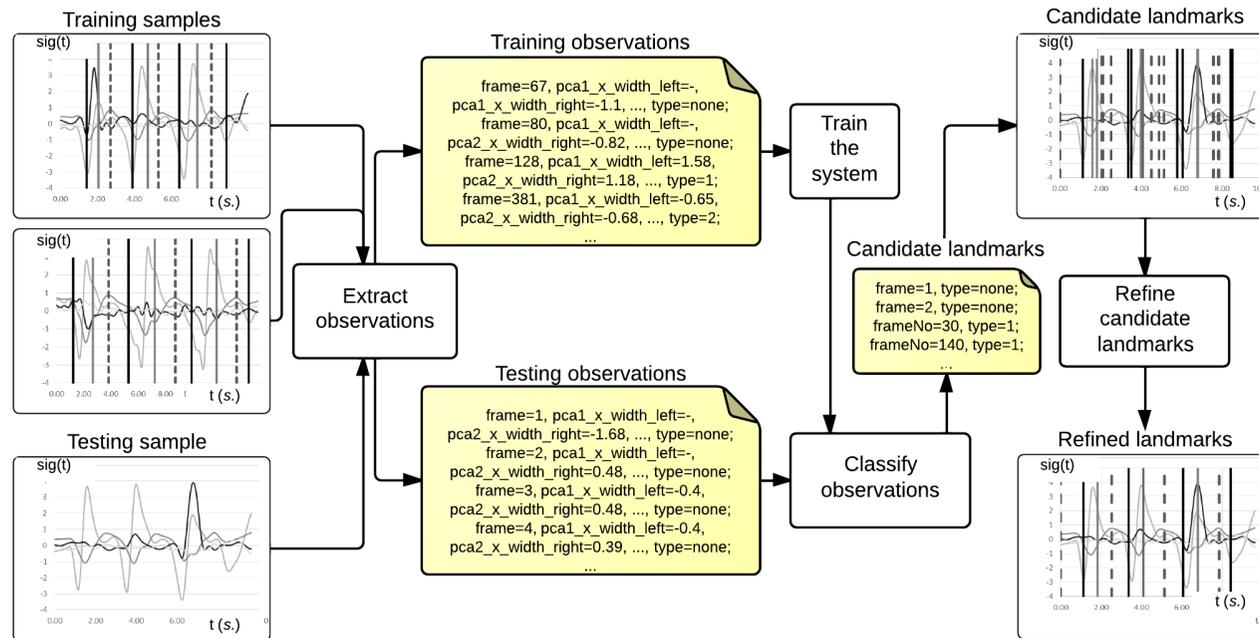
Several clustering methods can be used to tackle the first subtask:

- $k$ -means clustering [103] is based on the partitioning of samples into landmark clusters so that observations with similar mean feature values are grouped together. The Weka implementation of  $k$ -means clustering has been used to test this approach. If  $k(\chi_i)$  is the function that assigns a cluster number to a landmark, then  $\chi_i \in \sigma_{k(\chi_i)} \forall \chi_i \in \chi$
- The Expectation-Maximisation ( $EM$ ) algorithm [34] iteratively computes maximum likelihood estimates from incomplete data. The Weka implementation of  $EM$  has been used to test this approach. Let  $\sigma_{EM(\chi_i)}$  be the landmark cluster assigned to a landmark  $\chi_i$ , so that  $\chi_i \in \sigma_{EM(\chi_i)} \forall \chi_i \in \chi$ .
- Clustering by proximity in time of the landmarks. The previous algorithms do not take into account the temporal information inherent on each observation of  $\chi$ . Let us assume that  $\chi$  is ordered by  $FRAME(\chi)$ . Let  $FRAME(\chi_i) - FRAME(\chi_{i-1})$  be the distance in time between two contiguous landmarks  $\chi_i$  and  $\chi_{i-1}$ . When this value is smaller by some  $\beta \in \mathbb{R}$  than the time difference between the two previous pair of contiguous landmarks  $\chi_{i-1}$  and  $\chi_{i-2}$ , one can intuitively consider all  $\chi_i$ ,  $\chi_{i-1}$  and  $\chi_{i-2}$  belonging to the same cluster. In other words, lets consider the recursive function  $T_c(\chi_i)$  as

$$T_c(\chi_i) = \begin{cases} 1, & \text{if } i = 1. \\ T_c(\chi_{i-1}) + 1, & \text{if } i > 2 \text{ and } \frac{FRAME(\chi_i) - FRAME(\chi_{i-1})}{FRAME(\chi_{i-1}) - FRAME(\chi_{i-2})} > \beta. \\ T_c(\chi_{i-1}), & \text{otherwise.} \end{cases} \quad (5.29)$$

$T_c(\chi_i)$  will, this way, assign a unique landmark cluster identifier for each landmark, so that  $\chi_i \in \sigma_{T_c(\chi_i)}$ .

FIGURE 5.6: Illustration of landmark identification and landmark refining stages. Observations ( $\mathcal{O}(i)$ ) are represented by means of Weka *instance features* and their values, as well as the frame number and the observation class label (which is initially set to “none” for the testing observations)



### 5.5.2 Choosing the most suitable candidate landmark from each landmark cluster

Within each identified landmark cluster  $\sigma_r$ , an observation has to be chosen as a final refined landmark. Two different approaches have been considered:

- Selecting the candidate landmark whose feature values are closest to the mean feature values of all ground truth landmarks  $\chi^G$  of the same type of all training samples. Given a motion data series represented by Euler angles  $E$ , a landmark sequence  $\bar{\chi}$  and a landmark type  $\tau$ , let  $\bar{P}$  be the *mean pose per landmark type*, calculated as

$$\bar{P}(M, \chi, \tau) = \sum_{\chi_i \in L(\chi, \tau)} M(\text{FRAME}(\chi_i)) / |L(\chi, \tau)|, \quad (5.30)$$

where  $L(\chi, \tau) = \{\chi_i \in \chi / \text{TYPE}(\chi_i) = \tau\} \forall \tau \in \text{TYPE}(\chi)$ . This is the average body pose of a performance resulting on all landmark time frames of landmark type  $\tau$ . Then, the function  $\mu_1$  for landmark cluster selection on a landmark cluster  $\sigma_r$  is defined as

$$\begin{aligned} \mu_1(\sigma_r, sig, \chi^G, sig_T) = \{ \arg \min_{\sigma_r(h)} \text{Dist} ( \\ sig_i(\text{FRAME}(\sigma_r(h)), \bar{P}(sig_T, \chi^G, \text{TYPE}(\sigma_r(h))) \\ ) \} \forall \sigma_r \in \sigma, \end{aligned} \quad (5.31)$$

with  $sig_T$  being the motion signature of the tutor. The Euclidean distance between features is used to measure the affinity between two different observations. In this case, the features used to compare each candidate landmark is the value of the motion signature.

- Selecting the candidate landmark whose landmark type has the highest frequency within the landmark cluster and whose feature values are closest to the mean feature values of all ground truth landmarks of the same landmark type. That is,

$$\begin{aligned} \mu_2(\sigma_r, sig, \chi^G, sig_T) = \{ \arg \min_{\sigma_r(h)} \text{Dist} ( \\ sig(\text{FRAME}(\sigma_r(h)), \bar{P}(sig_T, \chi^G, \Xi(\sigma_r)) \\ ) \} \forall \sigma_r \in \sigma, \end{aligned} \quad (5.32)$$

where

$$\Xi(\sigma_r) = \arg \max_{\tau} |\{\chi_i \in \sigma_r / \text{TYPE}(\chi_i) = \tau\}| \forall \tau \in \text{TYPE}(\sigma_r) \quad (5.33)$$

## 5.6 Experimental results

In this chapter, results from a series of experiments that have been carried out in order to compare the performance of the proposed method against existing methods are presented. The involved exercises are summarised in Table 5.1. An Intel Core i5 3.10GHz PC with 4GB Random-Access Memory (RAM) was used to conduct the evaluation.

TABLE 5.1: Details of the features of the training and testing samples for landmark identification and landmark refining, with  $T$ =Number of landmark types (number of *classes*),  $G$ =average number of ground truth landmarks per sample,  $Ts$ =number of training samples per exercise,  $\overline{Tr}$ =average number of training frames per sample,  $\overline{Ts}$ =average number of testing frames per sample,  $M^P$ =motion data source (where P=absolute positions and Q=quaternions),  $\omega$ =resampling window length (in *s.*),  $N_j$ =number of involved joints and  $\beta$ =landmark cluster threshold (in *frames*)

Exercise	$T$	$G$	$Ts$	$\overline{Tr}$	$\overline{Ts}$	$M^P$	$\omega$	$N_j$	$\beta$
<i>Ankles</i> stretches	4	14	49	54	586	P	0.40	4	15
<i>Arms</i> raises	2	7	52	27	413	P	0.50	4	12
<i>Calves</i> stretches	4	14	41	55	638	P	0.50	4	15
<i>Inner thighs</i> stretches	2	7	51	27	340	Q	0.50	5	12
<i>Shoulders</i> and upper back stretches	3	9	49	37	391	P	0.50	4	12
<b>Average</b>	<b>3</b>	<b>10</b>	<b>48</b>	<b>40</b>	<b>474</b>				

Table 5.2 shows the selected body joints per exercise that have been selected to calculate the features from.

TABLE 5.2: Involved body joints per exercise

Exercise	Involved body joints
<i>Ankles</i>	LEFT_KNEE, LEFT_FOOT, RIGHT_KNEE, RIGHT_FOOT
<i>Arms</i>	LEFT_SHOULDER, LEFT_ELBOW, RIGHT_SHOULDER, RIGHT_ELBOW
<i>Calves</i>	LEFT_HIP, LEFT_KNEE, RIGHT_HIP, RIGHT_KNEE
<i>Inner thighs</i>	HIP, LEFT_HIP, LEFT_KNEE, RIGHT_HIP, RIGHT_KNEE
<i>Shoulders</i>	LEFT_ELBOW, RIGHT_ELBOW, LEFT_SHOULDER, RIGHT_SHOULDER

Ten individuals, whose age, sex, height and weight data can be seen in Table 5.3, executed a series of performances of each motion class and were asked to change the execution by varying the speed and the extent of the stretch. The quality of execution was manually assessed and annotated, in order to evaluate the sensibility of the proposed approach towards variations in the performances.

TABLE 5.3: Bio-information of the participants

ID	Age (y.)	Sex (M/F)	Height (cm.)	Weight (kg.)
P0	32	M	174	91
P1	30	M	178	85
P2	27	M	180	77
P3	25	M	183	74
P4	28	M	177	90
P5	69	F	152	-
P6	31	F	157	57
P7	32	M	174	82
P8	36	M	182	83
P9	56	M	168	65
<b>Average</b>	<b>37</b>	<b>-</b>	<b>173</b>	<b>78</b>
<b>St. dev.</b>	<b>14</b>	<b>-</b>	<b>11</b>	<b>11</b>
<b>Median</b>	<b>32</b>	<b>-</b>	<b>176</b>	<b>82</b>

### 5.6.1 Evaluation measures

Each exercise sample was manually labelled, according to stretches limit criteria, producing a different ground truth landmarks set  $\chi^G$  for each of them. A series of frames may represent a key body pose (extensively due to transitional phases), rather than just a specific time-frame. For this reason, a compromise interval  $[\Lambda^0, \Lambda^1]$  has been established for the ground truth landmarks, so that  $\chi^G = \{\{\text{FRAME}(t), \text{TYPE}(t), \Lambda_t^0, \Lambda_t^1\}\}$ . The compromise intervals are only considered for accuracy and precision measurement purposes, not to train the system.

The compromise intervals of each ground truth landmark is calculated automatically before any proficiency evaluation is done. For a given ground truth landmark  $\chi_i \in \chi^G = \{\chi_1.. \chi_n\}$  and a *difference threshold*  $\Delta_\Lambda \in [0, 1]$  of a motion data series  $M(t, j_i) = \{M_1..M_T\}$ , the interval limits  $\Lambda^0$  and  $\Lambda^1$  are calculated as

$$\begin{aligned}
\Lambda^0 &= \min \{ \Lambda \in [\text{FRAME}(\chi_{i-1}), \text{FRAME}(\chi_i)] / \\
&\quad \|M_t - M_{\text{FRAME}(\chi_i)}\|_1 \leq \|M_{\text{FRAME}(\chi_i)} - M_{\text{FRAME}(\chi_{i-1})}\|_1 \cdot \Delta_\Lambda \\
&\quad \forall t \in [\Lambda, \text{FRAME}(\chi_i)] \} \\
\Lambda^1 &= \max \{ \Lambda \in [\text{FRAME}(\chi_i), \text{FRAME}(\chi_{i+1})] / \\
&\quad \|M_t - M_{\text{FRAME}(\chi_i)}\|_1 \leq \|M_{\text{FRAME}(\chi_i)} - M_{\text{FRAME}(\chi_{i+1})}\|_1 \cdot \Delta_\Lambda \\
&\quad \forall t \in [\text{FRAME}(\chi_i), \Lambda] \},
\end{aligned} \tag{5.34}$$

where  $\|\cdot\|_1$  is the Taxicab norm,  $\text{FRAME}(\chi_0) = 1$  and  $\text{FRAME}(\chi_{n+1}) = \mathbb{T}$ . Examples of compromise intervals automatic calculation can be seen in Figure 5.7.

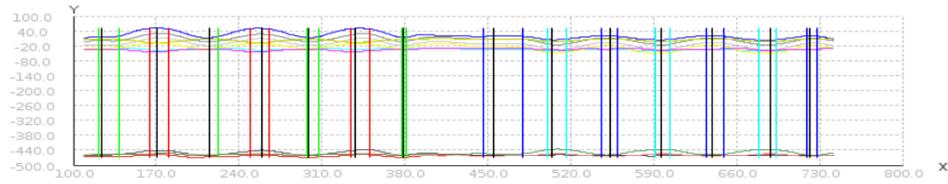
*Precision* ( $\pi_P$ ) and *accuracy* ( $\alpha_P$ ) of identified landmarks were chosen to measure the overall performance of every experiment run. Additionally, *false positive* ( $fp_P$ ), *false negative* ( $fn_P$ ), *true positive* ( $tp_P$ ) and *true negative* ( $tn_P$ ) of identified landmarks rates can be used as a complimentary measure of efficiency. These are calculated as

$$\begin{aligned}
tp_P(\bar{\chi}, \chi^G) &= |\{ \chi_i \in \bar{\chi} / \exists \chi_j \in \chi^G / \text{FRAME}(\chi_i) \in [\Lambda_j^0, \Lambda_j^1] \\
&\quad \text{AND TYPE}(\chi_i) = \text{TYPE}(\chi_j) \}|, \\
fp_P(\bar{\chi}, \chi^G) &= |\{ \chi_i \in \bar{\chi} / \forall \chi_j \in \chi^G \text{ FRAME}(\chi_i) \notin [\Lambda_j^0, \Lambda_j^1] \\
&\quad \text{OR TYPE}(\chi_i) \neq \text{TYPE}(\chi_j) \}|, \\
fn_P(\bar{\chi}, \chi^G) &= |\{ \chi_j \in \chi^G / \forall \chi_i \in \bar{\chi} \text{ FRAME}(\chi_i) \notin [\Lambda_i^0, \Lambda_i^1] \\
&\quad \text{OR TYPE}(\chi_i) \neq \text{TYPE}(\chi_j) \}|, \\
tn_P(\bar{\chi}, \chi^G) &= |\bar{\chi}| - tp_P(\bar{\chi}, \chi^G) - fp_P(\bar{\chi}, \chi^G) - fn_P(\bar{\chi}, \chi^G).
\end{aligned} \tag{5.35}$$

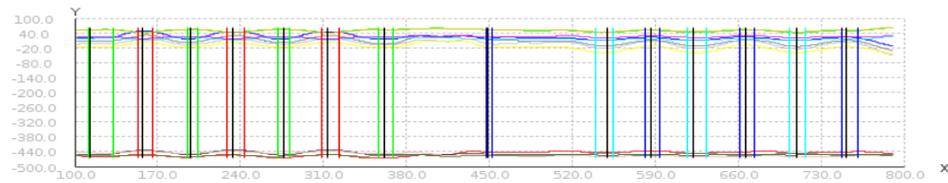
Thus, given a set of refined landmarks  $\bar{\chi}$ ,  $\pi_P$  and  $\alpha_P$  are calculated as

$$\begin{aligned}
\pi_P(\bar{\chi}, \chi^G) &= tp_P(\bar{\chi}, \chi^G) / (tp_P(\bar{\chi}, \chi^G) + fp_P(\bar{\chi}, \chi^G)) \\
\alpha_P(\bar{\chi}, \chi^G) &= (tp_P(\bar{\chi}, \chi^G) + tn_P(\bar{\chi}, \chi^G)) / \\
&\quad (tp_P(\bar{\chi}, \chi^G) + tn_P(\bar{\chi}, \chi^G) + fp_P(\bar{\chi}, \chi^G) + fn_P(\bar{\chi}, \chi^G)).
\end{aligned} \tag{5.36}$$

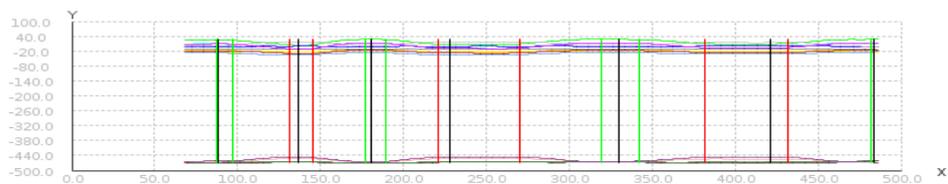
FIGURE 5.7: Example of compromise intervals calculated automatically with  $\Delta_{\Lambda} = 0.15$ . Time in the X axis (in *s.*) and absolute body joint/DOF positions in the Y axis (in *cm.*). Each group of three vertical marker denotes a ground truth landmark and its compromise interval. The marker colour of the interval limits represents the landmark type, while the manually labelled unique frame value is represented in black



(a) Ankle stretches



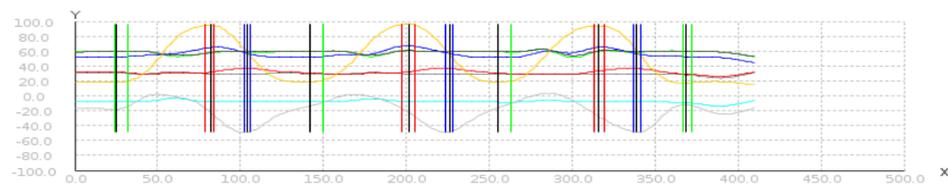
(b) Calf stretches



(c) Inner thigh stretches



(d) Arm raises



(e) Shoulder stretches

## 5.6.2 Comparison of different learning and clustering techniques for landmark identification

Precision and accuracy rates achieved by the three classifiers described in Section 5.4.4 and based on a 10-fold cross-validation (5 repetitions) of landmark identification are compared in Table 5.4.

The efficacy of AdaBoost is tested by comparing the results of these machine learning classifiers alone with those obtained when AdaBoost is used as well. Both training times (per training set) and execution time (per sample) are also shown.

TABLE 5.4: Comparison of average precision ( $\pi_P$ ) and accuracy ( $\alpha_P$ ) for landmark identification using different machine learning classifiers, where  $AB?$ =whether AdaBoost is used along the machine learning classifier and  $TT$ ,  $ET$ =average training and execution times, respectively (in  $s$ )

Machine learning classifier	$AB?$	$\pi_P$	$\alpha_P$	$TT$	$ET$
<b>C4.5</b>	no	0.20	0.52	0.0376	0.0009
	yes	0.28	0.60	0.3106	0.0106
<b>Naive Bayes</b>	no	0.17	0.56	0.0245	0.0190
	yes	0.18	0.57	0.4487	0.2652
<b>Support vector machines</b>	no	0.52	0.45	0.0823	0.0012
	yes	0.22	0.58	1.1414	0.0168

The results show that, apart from very poor numbers for precision and accuracy, support vector machines are seemingly superior to the others, especially when AdaBoost is *NOT* used. This may be due to the fact that AdaBoost is designed to improve the results of WEAK classifiers, whereas support vector machines are considered as STRONG classifiers. AdaBoost may thus play an adverse role in combination with support vector machines.

Also, AdaBoost contributes towards higher training and execution times, although these are always smaller than a second. The low precision and accuracy achieved by these algorithms is caused by the high rate of redundant false positives, i.e., the identification of multiple contiguous landmarks. Since only one matched landmark with the ground truth landmarks is considered a true positive, the rest of matched landmarks are considered as false positives, although they are of the same landmark type as the matched ground truth landmark. This situation is tackled by the landmarks clustering and landmark cluster selection methods.

Table 5.5 shows a comparison of the proficiency achieved by combined landmark identification and landmark refining approaches. The identified landmarks given by each landmark identification algorithm are refined by means of clustering techniques (EM,  $k$ -means and  $T_c$ ) and landmark cluster selection algorithms ( $\mu_1$  and  $\mu_2$ ).

The results show that much higher precision and accuracy can be achieved by using either EM or  $k$ -means clustering and  $\mu_1$  for landmark clustering and landmark cluster selection, respectively.

TABLE 5.5: Comparison of average precision ( $\pi_P$ ) and accuracy ( $\alpha_P$ ) per testing sample for landmark identification+landmark refining using different algorithms for landmark identification, clustering and landmark cluster selection, where *Clust.*=Clustering algorithm, *Sel.*=Selection algorithm, *+AB*=with AdaBoost,  $T_c$ =temporal proximity landmark clustering algorithm, and  $\mu_1, \mu_2$ =landmark cluster selection algorithms (see Section 5.5)

Clust. Sel.	C4.5		(+AB)		Naive Bayes		(+AB)		Support vector machines		(+AB)	
	$\pi_P$	$\alpha_P$	$(\pi_P$	$\alpha_P)$	$\pi_P$	$\alpha_P$	$(\pi_P$	$\alpha_P)$	$\pi_P$	$\alpha_P$	$(\pi_P$	$\alpha_P)$
EM	$\mu_1$	0.67	0.89	<b>(0.72</b>	<b>0.88)</b>	0.64	0.92	(0.65	0.91)	0.70		
	$\mu_2$	0.59	0.87	(0.66	0.86)	0.60	0.92	(0.61	0.91)	0.69	0.54	(0.66 0.89)
$k$	$\mu_1$	0.67	0.89	(0.72	0.87)	0.64	0.92	(0.65	0.91)	0.70	0.54	(0.65 0.89)
	$\mu_2$	0.58	0.87	(0.65	0.86)	0.60	0.92	(0.61	0.90)	0.71	0.56	(0.65 0.89)
$T_c$	$\mu_1$	0.66	0.89	(0.70	0.88)	0.64	0.92	(0.64	0.91)	0.71	0.56	(0.65 0.89)
	$\mu_2$	0.58	0.87	(0.65	0.86)	0.60	0.92	(0.61	0.91)	0.69	0.55	(0.63 0.88)

Additionally, C4.5 and AdaBoost for landmark identification show to perform better than support vector machines when combined with the clustering methods above. Figure 5.8 shows a graphical comparison of a number of landmarks identified by C4.5 and support vector machines ( $\chi_C$  and  $\chi_S$  respectively) and the refined landmark using EM and  $\mu_1$  on both ( $\bar{\chi}_C$  and  $\bar{\chi}_S$  respectively)

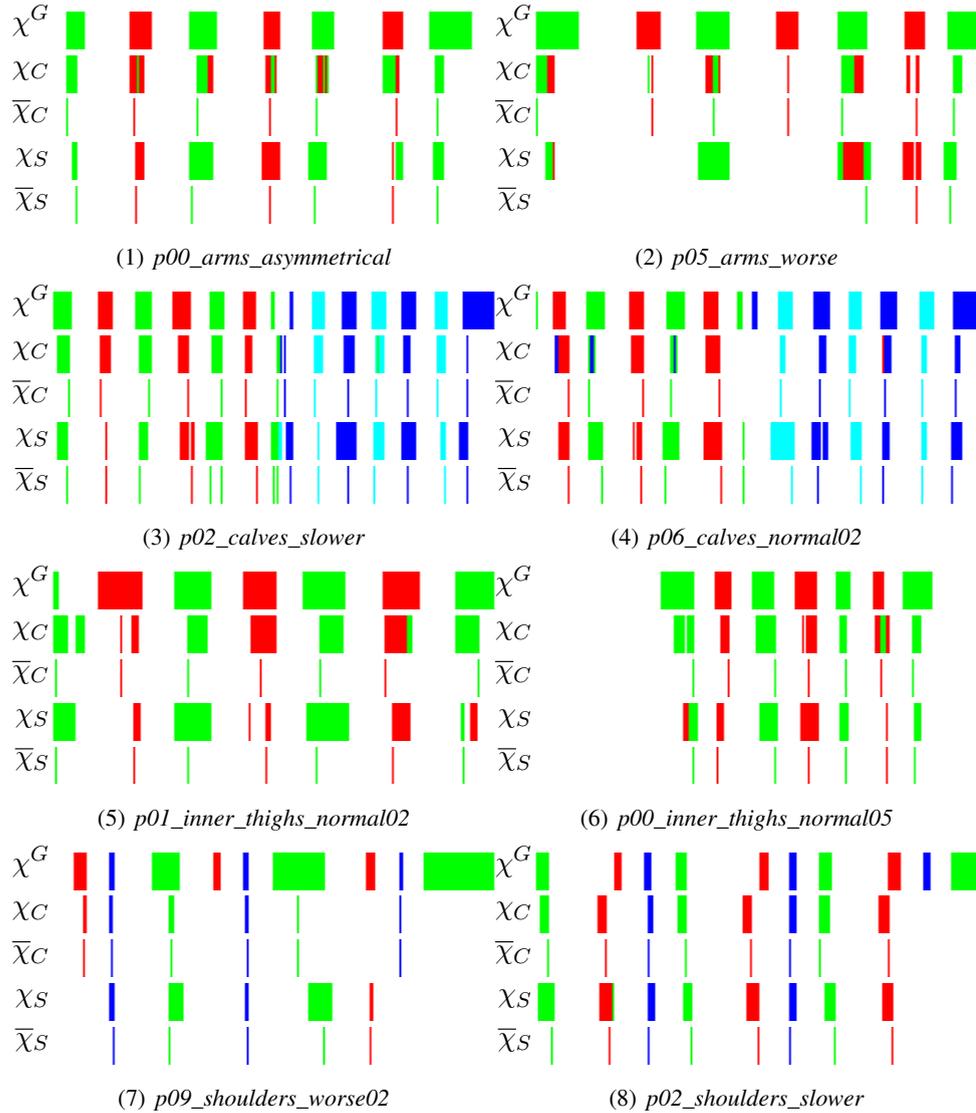
Figure 5.8 suggests that support vector machines are less precise than C4.5 and identified a broader range of body poses in the vicinity of each ground truth landmark. Also, support vector machines fail to identify some landmarks.

The landmark refining stage shows an obvious improvement for C4.5+AdaBoost, where the precision achieved (72%) is over 2.5 times higher than the one achieved for landmark identification only (28%), while the accuracy (87%) is also significantly higher (than 60%).

Finally, the average execution time of each clustering and landmark cluster selection method is shown in Table 5.6. This execution time can be considered as insignificant (between 1 and 10 *ms.*).

Experiments involving the use of the motion signature (see Section 5.2) and the frame-wise features (see Section 5.4.2) were performed. When not using the motion signature, the smoothed input data was used instead. When not extracting frame-wise features, either the motion signature or the original smoothed data was used. To help understand better the influence of the parameters,

FIGURE 5.8: Comparison of results for landmark identification and landmark refining of C4.5 and support vector machines. Ground truth landmarks ( $\chi^G$ ), identified landmarks using C4.5 ( $\chi_C$ ) and support vector machines ( $\chi_S$ ) and the respective refined landmarks ( $\bar{\chi}_C$  and  $\bar{\chi}_S$ ) represented with vertical markers of different colours, each of which represent a ground truth, identified or refined landmark type, respectively



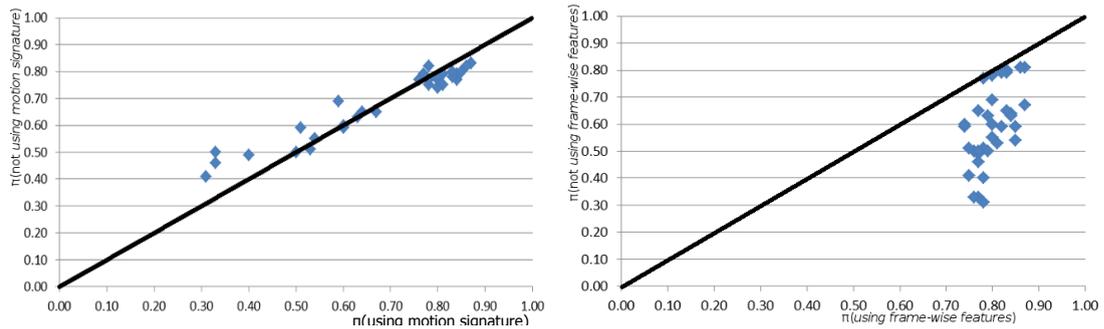
a scatter plot of the precision achieved is shown in Figure 5.9, with different values for the “Use motion signature” (a) and “Extract frame-wise features” (b).

The figure reveals that the use of motion signature does not significantly improve the average performance. It is worth noting that only the motion data of the involved body joints for each motion class was used, hence the small contribution of the dimensionality reduction technique. On the other hand, the frame-wise features do contribute to a better overall performance, probably due to its capacity for modelling discrete multi-variate data from the point of view of the differences between neighbouring points.

TABLE 5.6: Comparison of average execution time for landmark refining using different algorithms, where  $T_c$ =temporal proximity landmark clustering algorithm and  $\mu_1, \mu_2$ =landmark cluster selection algorithms (see Section 5.5)

Clustering Algorithm	Selection Algorithm	Execution time (in $\mu s$ )
$EM$	$\mu_1$	6673.02
	$\mu_2$	6673.76
$k$	$\mu_1$	1580.26
	$\mu_2$	1588.64
$T_c$	$\mu_1$	1045.71
	$\mu_2$	1055.21

FIGURE 5.9: Comparison of average precision achieved for landmark identification+landmark refining per testing sample using/not using the motion signature (a) and extracting/not extracting frame-wise features (b)



(a)  $p(x,y)=\pi(\text{using motion signature}), \pi(\text{using original data})$  (b)  $p(x,y)=\pi(\text{extracting frame-wise features}), \pi(\text{original data})$

For the parameters with which the method achieved the best precision rate (AdaBoost + C4.5 + EM +  $\mu_1$ ), Table 5.7 gives further details of accuracy, false negatives and positives rates for both the landmark identification and landmark refining stages. The results are shown in relation to the five involved exercises.

These results show that both precision and accuracy vary across motion classes. The *Shoulders* exercise showed to be the one with both the lowest precision and accuracy rates. A graphical visualisation of a number of candidate landmarks and refined landmarks for a series of individuals on different motion classes are shown in Figure 5.10. Further examples can be found in Appendix A.

TABLE 5.7: Average precision ( $\pi$ ), accuracy ( $\alpha$ ), false positives rate ( $fp$ ) and false negatives rate ( $fn$ ) achieved for landmark identification and landmark refining (clustering and landmark cluster selection) per testing sample, with  $PT$  and  $TT$  = Pre-processing and training times, respectively (in  $s$ ). 10-fold cross-validation with 5 repetitions

Exercise	$PT$	Landmark identification ( $\chi$ )					Landmark refining ( $\bar{\chi}$ )			
		$\pi_P$	$\alpha_P$	$fp_P$	$fn_P$	$TT$	$\pi_P$	$\alpha_P$	$fp_P$	$fn_P$
Ankles	3.42	0.30	0.52	27.29	4.71	0.61	0.67	0.80	3.42	6.34
Arms	2.62	0.23	0.61	29.11	1.02	0.14	0.77	0.93	1.32	1.85
Calves	3.33	0.31	0.61	28.04	3.86	0.46	0.73	0.87	2.80	5.48
Inner thighs	2.20	0.32	0.74	18.27	1.05	0.19	0.83	0.95	0.96	1.82
Shoulders	2.21	0.25	0.53	24.66	2.73	0.15	0.60	0.83	3.11	4.39
<b>Average</b>	<b>2.73</b>	<b>0.28</b>	<b>0.60</b>	<b>25.37</b>	<b>2.6</b>	<b>0.30</b>	<b>0.72</b>	<b>0.92</b>	<b>2.29</b>	<b>3.88</b>

This illustrates the effect of landmark identification and the similarity of adjacent poses, as well as the effectiveness of the landmark clustering and selection methods.

### 5.6.3 Comparison with Dynamic Time Warping (DTW) and Hierarchical Aligned Cluster Analysis (HACA)

TABLE 5.8: Comparison of average precision ( $\pi$ ), accuracy ( $\alpha$ ), false positives rate ( $fp$ ) and false negatives rate ( $fn$ ) for landmark identification per testing sample using the proposed approach, DTW and HACA

Exercise	C4.5+AdaBoost +EM+ $\mu_1$				DTW				HACA		
	$\pi_P$	$\alpha_P$	$fp_P$	$fn_P$	$\pi_P$	$\alpha_P$	$fp_P$	$fn_P$	$\pi_P$	$fp_P$	$fn_P$
Ankles	0.67	0.80	3.42	6.34	0.39	0.27	7.58	8.50	0.11	2.93	4.67
Arms	0.77	0.93	1.32	1.85	0.60	0.51	2.61	2.47	0.04	5.71	11.00
Calves	0.73	0.87	2.80	5.48	0.52	0.40	6.28	6.65	0.05	5.85	10.33
Inner thighs	0.83	0.95	0.96	1.82	0.83	0.74	1.20	1.06	0.11	3.27	4.65
Shoulders	0.60	0.83	3.11	4.39	0.39	0.27	6.02	5.44	0.38	0.18	4.37
<b>Average</b>	<b>0.72</b>	<b>0.92</b>	<b>2.29</b>	<b>3.88</b>	<b>0.55</b>	<b>0.44</b>	<b>4.62</b>	<b>4.70</b>	<b>0.10</b>	<b>4.43</b>	<b>7.02</b>

Table 5.8 shows the average precision, false positive and false negative rates achieved on landmark refining for all the samples of each motion class using the proposed method and the DTW [8] and HACA [127] methods (see Section 2.2.1 and Section 2.2.2, respectively). Average execution and training times (when applicable) are shown in Table 5.9.

FIGURE 5.10: Representation of results for landmark identification and landmark refining. Ground truth landmarks ( $\chi^G$ ), candidate landmarks ( $\chi$ ) and refined landmarks ( $\bar{\chi}$ ) represented with vertical markers of different colours, each of which representing a ground-truth, identified or refined landmark type, respectively

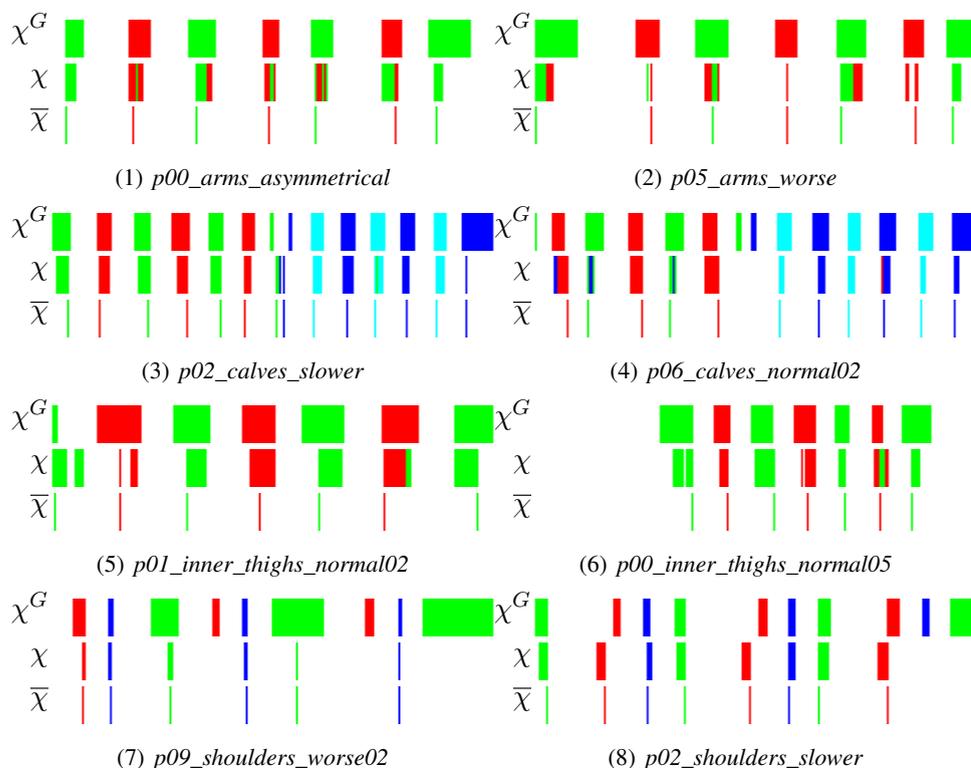


Figure 5.11 shows a graphical representation of the results achieved for landmark identification and landmark refining, comparing the performance of both cluster analysis and DTW.

Some samples have different levels of landmark sequence repetitiveness and landmark sequence symmetry. Ground truth landmarks ( $\chi^G$ ) are represented by boxes, spanning across the compromise range, with each different marker colour representing a landmark type, while refined landmarks ( $\bar{\chi}$ ) are represented by vertical lines of the corresponding marker colour. Specifically, Figure 5.11 (1) shows how DTW fails to align samples with a smaller degree of landmark sequence repetitiveness, i.e., exceeding stretch repetitions in the training samples are not correctly matched. Additionally, the effect of landmark sequence symmetry is shown in Figure 5.11 (5) on a testing sample in which the user exercised the right side of the body first, as opposed to the left side first.

TABLE 5.9: Average on-line and off-line execution time per testing sample for landmark identification using several different methods, with  $PT$ ,  $TT$ ,  $ET$ =pre-processing, training and execution times, respectively, and  $TOT=PT + ET$  (in  $s.$ ). Note that C4.5+AdaBoost+EM+ $\mu_1$  use frame-wise features calculated from the motion signature, while DTW and HACA use the raw motion data

Exercise	$PT$	C4.5			$PT$	DTW		HACA	
		+AdaBoost				$ET$	$TOT$	$ET$	$TOT$
		+ EM + $\mu_1$							
		$TT$	$ET$	$TOT$					
Ankles	3.42	0.94	0.03	4.39	0.79	0.68	1.47	14.10	14.89
Arms	2.62	0.21	0.01	2.84	0.58	0.43	1.01	24.86	25.44
Calves	3.33	1.02	0.03	4.38	0.87	0.79	1.66	18.56	19.43
Inner thighs	2.20	0.30	0.01	2.51	0.57	0.52	1.09	9.60	10.17
Shoulders	2.21	0.25	0.01	2.47	0.51	0.23	0.74	12.49	13.00
<b>Average</b>	<b>2.73</b>	<b>0.46</b>	<b>0.02</b>	<b>3.32</b>	<b>0.65</b>	<b>0.52</b>	<b>1.17</b>	<b>15.92</b>	<b>16.59</b>

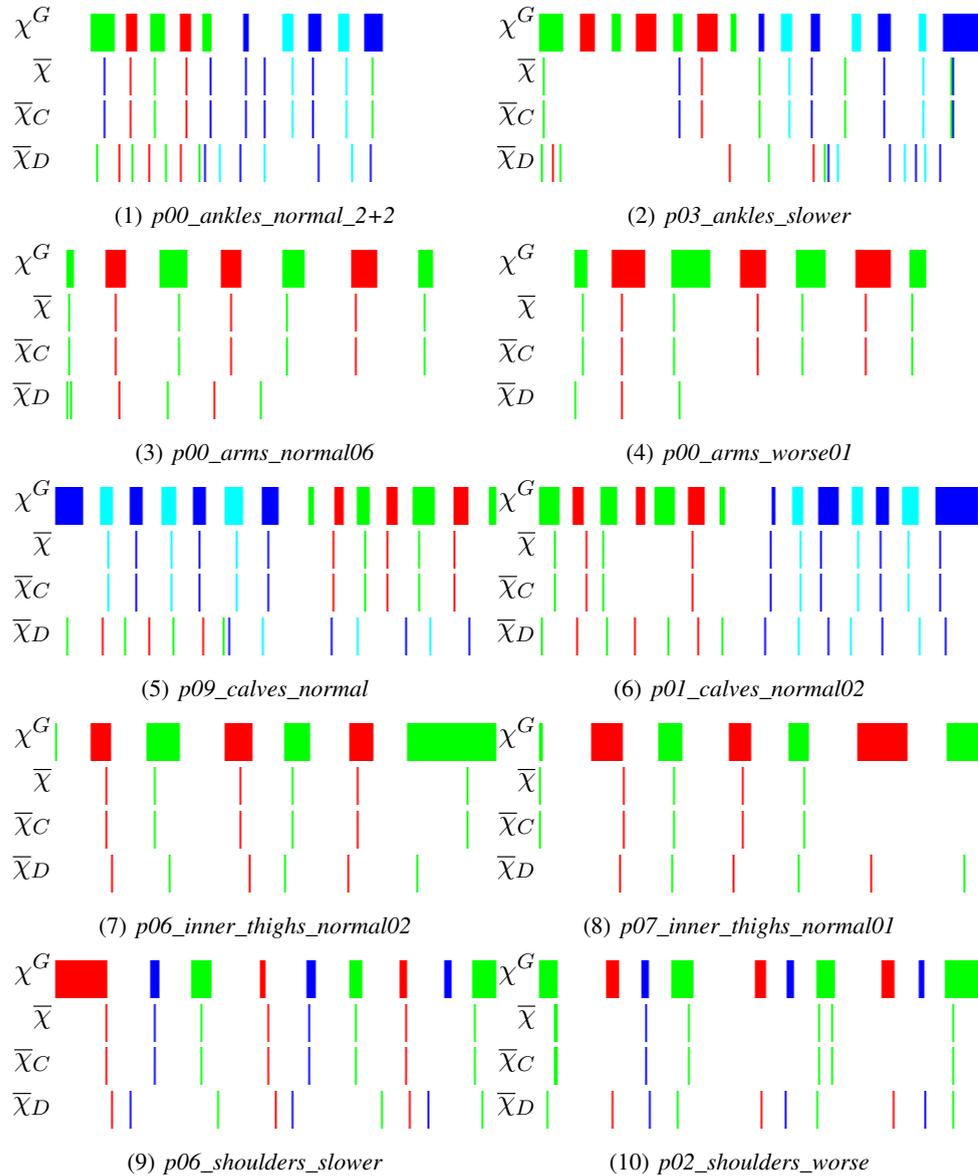
## 5.7 Conclusions

In this chapter, a novel semi-supervised method for key body pose (or landmark) identification for a given set of motion data samples has been introduced. The techniques used to achieve this include well-known machine learning algorithms operating over series of observations extracted from frame-wise information of the data source. Each of these observations are classified into pre-defined landmark types. The method thus not only recognises key body poses, but also infers the kind of pose within a previously-known set of learnt poses.

The proposed approach has a number of advantages which include versatility in terms of landmark sequence repetitiveness and landmark sequence symmetry of the data. Furthermore, the proposed approach takes advantage of an *ad-hoc* frame-wise feature generation function that models the relation between significant and noise-related data. Section 5.6 shows a series of results drawn from an empirical evaluation of the proposed approach using real data and a comparison with alternative methods for motion segmentation and alignment.

Several machine learning algorithms have been compared and evaluated. A series of methods for landmark clustering and landmark cluster selection have also been compared and evaluated, in order to improve the results of the proposed approach. The results show that a ground

FIGURE 5.11: Comparison of results between landmark identification+landmark refining and DTW. Ground truth landmarks ( $\chi^G$ ), refined landmarks using a machine learning classifier+cluster analysis ( $\bar{\chi}_C$ ) and DTW ( $\bar{\chi}_D$ ) represented with vertical markers of different colours, each of which representing a ground-truth, refined or identified landmark type, respectively



truth landmark match average 72% precision and 87% accuracy rates are achieved, including a compromise interval criteria.

The proposed approach has been compared with two motion alignment and clustering algorithms: HACA and DTW. The former is not sufficiently accurate and is inappropriate for the application. This shows that methods dealing with the whole spectrum of motion data do not suit the needs of exact momenta identification. These methods generate too many false positives and false negatives.

DTW shows a high sensitivity to landmark sequence repetitiveness and landmark sequence symmetry of the motion data. If two sequences belong to the same motion class but one of them differs in the amount of stretch repetitions, the warping path will be incorrect. Furthermore, if the exercise requires the individual to execute the primitives in a symmetric manner (i.e. stretch the left calf first, then the right one) and the user exercises the opposite side first, the result will be inconsistent. Results on Landmark identification using the two evaluated methods have been discussed.

Additionally, the proposed approach has a low on-line mean execution time of 3.32 s per sample (including pre-processing, i.e., motion signature calculation and frame-wise features extraction), with prior training taking place off-line. This is a very important feature in order to implement this methodology in real-time (RT). Future usable implementations of the approach in real scenarios may take advantage of in-line calculations, given the frame-wise nature of the machine learning classifier, i.e., classification of a single pose depends only on the adjacent poses, in order to calculate the frame-wise features, as explained in Section 5.4.2.

## Chapter 6

# A genetic algorithm for periodic landmark sequence analysis based on landmark sequence fitness optimisation

In this chapter, the problem of having a periodic sequence of string elements which contains out-of-sequence false positives (i.e., elements with types that do not fit the pattern within the sequence) is addressed.

The problem of inexact or approximate string matching is present in many different applications: from spell checking [111] to ADN sequence matching [6] to spam filtering. The application addressed in this paper is the analysis of periodic key body poses and long periodic string sequences analysis.

The output of the landmark identification and landmark refining stages is an ordered sequence of candidate body poses. One of the issues with the approach is the precision achieved, where out-of-sequence landmark types may be present. Furthermore, a number of landmarks may remain undetected. Summing up, the resulting sequence will likely contain both false positives (*fp*) and false negatives (*fn*).

Although many algorithms for string, sub-string and subsequence alignment is currently available [6, 27, 60, 116], the complexity of the problem indicates that a more flexible method is needed, as they do not handle very well the identification and removal of misplaced items (which may, in

turn, be present –or not– on the ground truth). This chapter presents an evolutionary computation approach to out-of-sequence member identification and removal for periodic sequences.

The theoretical background of the problem being addressed is firstly introduced. A Genetic Algorithm (GA) for extraneous member removal on periodic sequences is presented next. Then, results of its application in randomly generated string sequences and sequences of key body poses are shown. The approach is benchmarked with classical approximate and exact string matching techniques.

## 6.1 Theoretical background and problem formulation

Lets first define a sequence  $S$  as an ordered set of elements belonging to an alphabet  $\mathcal{A}$ . Thus,  $\mathcal{A}^*$  represents all possible sequences this way defined. e.g., if

$$\mathcal{A}_i = \{A, B, C, D, E, F\} \quad (6.1)$$

represents an alphabet, then the sequence

$$S_i = \{A, C, D, A, B, A, F, D, B\} \quad (6.2)$$

is a member of  $\mathcal{A}^*_i$ .

Let  $s_i$  be the  $i$ -th element of a given string  $S \in \mathcal{A}^*$ , so that  $S = \{s_1, s_2, s_3, \dots, s_N\}$ , being  $N = |S|$ . Subsequently, a *period*  $\nu = \{a_i, \dots, a_j\} \in \mathcal{A}$  is a special sequence where  $s_i = s_j$ .  $\nu^n$  is the concatenation of  $n$  instances of a given period  $\nu$ . Let  $\mathcal{N} \subset \mathcal{A}^*$  be the subset of all periods in a given alphabet  $\mathcal{A}$ . Finally, a *periodic sequence*  $S$  is defined as

$$S = \{\nu_1^{n_1} \nu_2^{n_2} \dots \nu_k^{n_k} / \nu_i \in \mathcal{N}\}. \quad (6.3)$$

To illustrate the problem, take the sequence of a given sample of an *Ankles stretch* exercise

$$\begin{aligned}
\chi^G = \{ & \{\text{FRAME} = t_1, \text{TYPE} = 1\}, \\
& \{\text{FRAME} = t_2, \text{TYPE} = 2\}, \\
& \{\text{FRAME} = t_3, \text{TYPE} = 1\}, \\
& \{\text{FRAME} = t_4, \text{TYPE} = 2\}, \\
& \{\text{FRAME} = t_5, \text{TYPE} = 1\}, \\
& \{\text{FRAME} = t_6, \text{TYPE} = 2\}, \\
& \{\text{FRAME} = t_7, \text{TYPE} = 1\}, \\
& \{\text{FRAME} = t_8, \text{TYPE} = 3\}, \\
& \{\text{FRAME} = t_9, \text{TYPE} = 4\}, \\
& \{\text{FRAME} = t_{10}, \text{TYPE} = 3\}, \\
& \{\text{FRAME} = t_{11}, \text{TYPE} = 4\}, \\
& \{\text{FRAME} = t_{12}, \text{TYPE} = 3\}, \\
& \{\text{FRAME} = t_{13}, \text{TYPE} = 4\}, \\
& \{\text{FRAME} = t_{14}, \text{TYPE} = 3\} \}
\end{aligned} \tag{6.4}$$

as an example. For clarity, let's redefine  $\chi^G$  as a sequence of key body pose types, i.e.,

$$\chi^G = \{\tau_1, \tau_2, \tau_1, \tau_2, \tau_1, \tau_2, \tau_1, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3\}, \tag{6.5}$$

where  $\tau_i$  stands for a key body pose of type  $i$ . This sequence represents a series of repetitions of the same *Ankle* stretches, first the left ankle ( $\tau_1$  and  $\tau_2$ ) and then the right ankle ( $\tau_3$  and  $\tau_4$ ). Let

$$\begin{aligned}
\text{TYPE}(\bar{\chi}) = \{ & \tau_1, \tau_2, \tau_1, \tau_1, \tau_2, \tau_3, \\
& \tau_1, \tau_2, \tau_1, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4 \}.
\end{aligned} \tag{6.6}$$

be a sequence of detected key body poses. One of either the third or the fourth member (both  $\tau_1$ ) does not belong there as per the ground truth  $\chi^G$ . Furthermore, the sixth member ( $\tau_3$ ) is clearly a false positive as it does not match the pattern (i.e., it represents a range of motion (RoM) edge

of the stretch of the right ankle within an unfinished stretch of the left ankle). Finally, the last member of  $\chi^G$  ( $\tau_3$ ) is missing, which is a false negative.

This is an example of common situations found in  $\bar{\chi}$ . In the case of false positives that fall within sub-strings of the same stretch (as in  $\{\dots, \tau_1, \tau_2, \tau_1, \tau_1, \tau_2, \dots\}$ ), there is no simple solution, since there is no way to know which of the elements in conflict is the correct one. However, the second case ( $\{\dots, \tau_2, \tau_3, \tau_1, \dots\}$ ) is different.  $\tau_3$  cannot be placed within a  $(\tau_1, \tau_2, \tau_1, \tau_2, \dots)$  sub-string. In this case, the *extraneous* element can be safely removed.

Thus, let  $\bar{\chi}^S \in \bar{\chi}$  be the set of key body poses after periodic sequence analysis by discovering and removing extraneous false positives.

Lets  $\mathcal{A}_\nu \subset \mathcal{A}$  be the sub-alphabet defined by  $\nu_i$ , so that  $\nu_i^* \subset \mathcal{A}_{\nu_i}$ . Lets consider the restriction of only using strings which sub-alphabets have null intersection, i.e.,

$$\begin{aligned} \bigcup_{i=1..k} \nu_i \in S &= \mathcal{A} \\ \bigcap_{i=1..k} \nu_i \in S &= \emptyset. \end{aligned} \tag{6.7}$$

Lets consider a periodic string  $S$  in (6.3), which periods are defined over null-intersection sub-alphabets of  $\mathcal{A}$ . Then, given one of its periods  $\nu_i(1) \in \nu_i^{n_i}$ ,

$$\nu_i(1)' = \{s_i, s_{i+1}, \dots, e, \dots, s_j\} \tag{6.8}$$

is a version of  $\nu_i(1)$  with an *extraneous* member  $e$ , so that  $\nu_i(1)' \neq \nu_i$ . When  $e \in \mathcal{A}_{\nu_i^*}$ ,  $e$  is an *in-period* extraneous member of  $\nu_i$ . If  $e \notin \mathcal{A}_{\nu_i^*}$ ,  $e$  is an *out-of-period* extraneous member.

In order to represent the ground truth, lets denote an in-period extraneous member  $s_i$  as  $s_i$ , an out-of-period extraneous member as  $(s_i)$  and a deleted member as  $[s_i]$ .

The remainder of this section introduces methods for sequence analysis and introduces a new method for *periodic* sequence analysis, which takes advantage of the cyclic nature of the sequences object of analysis.

## 6.2 Analysis of periodic sequences

The LCS (Largest Common Subsequence) problem [74] is perhaps the one that tackles more directly the task of removing out-of-sequence members in a sequence. The aim is to find the longest subsequence common to two sequences, with a subsequence being different to a substring, since the members of the latter must be consecutive in the original sequence. Thus, given the strings

$$\begin{aligned} S_1 &= AGCCTGATCCAGTTCTAACTTGACTG \\ S_2 &= AGCCGCAAGATGCAGTCATTAAGATGCTTG, \end{aligned} \quad (6.9)$$

the LCS to both will be

$$LCS(S_1, S_2) = AGCGATCAGTTTAAATGCTG. \quad (6.10)$$

Therefore, the LCS between  $\bar{\chi}$  and  $\chi^G$  (see (6.5) and (6.6)) will be

$$\begin{aligned} LCS(\chi^G, \bar{\chi}) &= \{\tau_1, \tau_2, \tau_1, \tau_2, \tau_1, \tau_2, \\ &\quad \tau_1, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4\}. \end{aligned} \quad (6.11)$$

However, this will fail, on some occasions, when applied to sequences with a repetitiveness effect, since there is no implicit consideration of the periodic nature of the sub-sequences. For example, if the sequence

$$\begin{aligned} \bar{\chi}' &= \{\tau_1, \tau_2, \tau_1, \tau_1, \tau_2, (\tau_3), \tau_1, \tau_2, \tau_1, \tau_2, \tau_1, \\ &\quad \tau_2, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4\}. \end{aligned} \quad (6.12)$$

is considered, in where, in addition to the extraneous members in  $\bar{\chi}$ , one more repetition is performed per side of the body (i.e., stretching the left ankle four times and stretching the right side four times), the LCS will be

$$LCS(\chi^G, \bar{\chi}') = \{\tau_1, \tau_2, [\tau_1], \tau_1, \tau_2, [(\tau_3)], \tau_1, \tau_2, \tau_1, [\tau_1], [\tau_2], \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3, \tau_4, \tau_3 [\tau_4]\}. \quad (6.13)$$

As a result, three true positives are lost. This is because these methods do not take into account the cyclic nature of the sub-sequences and, thus, do not work well with longer patterns.

### 6.2.1 Periodic analysis of a sequence

Let

$$S = \{s_1..s_N\}. \quad (6.14)$$

be a periodic string. Each periodic sub-string contained in it is defined as

$$\nu(S, i) = \{s_i, \dots, s_{\kappa(S, i)}\} \quad (6.15)$$

where

$$\kappa(S, i) = \min \{n = i + 1..N/s_n = s_i\}. \quad (6.16)$$

The value of the members placed at the beginning and end of a period ( $s_i$  and  $s_{\kappa(S, i)}$ , respectively) is the *heading* member. Let

$$\mathcal{N}(S) = \bigcup_{i=1..N} \nu(S, i) \quad (6.17)$$

be the set of all periods on  $S$ . Take the sequence

$$S = \{ABABABACDCDCDC\}, \quad (6.18)$$

belonging to the alphabet  $\mathcal{A} = \{A, B, C, D\}$ , as an example, where two type of periods can be clearly identified, i.e.

$$\mathcal{N}(S) = \{\{ABA\}, \{CDC\}\}. \quad (6.19)$$

### 6.2.2 Sequence fitness function

Given a series of *training patterns*  $\mathcal{P} = \{\bar{\nu}_1, \bar{\nu}_2, \dots, \bar{\nu}_n, \}$ , where  $\bar{\nu}_i \in \mathcal{P}$  is a period, and a given period  $\nu$  of a periodic sequence  $S$ , the sub-set of all the periods in  $\mathcal{P}$  that share the same heading with  $\nu$  is found as

$$\mathcal{M}(\nu, \mathcal{P}) = \bigcup \{\bar{\nu}_i \in \mathcal{P} / \bar{\nu}(1) = \nu(1)\}. \quad (6.20)$$

That is,  $\mathcal{M}$  is the stepping stone towards measurement of the landmark sequence period fitness of a given period  $\nu$ . The latter will be matched against all periods in the training patterns headed by the same member.

The landmark sequence period fitness of a given sequence  $S$  is calculated according to how well the periods match those that will typically appear on a common pattern. The objective is to produce sequences, based on the original sequence, with an improved fitness after each generation. This is done by eliminating members that do not fit the pattern (*out-of-period* and *in-period* extraneous members). With a high likelihood, these will be candidates of *false positives* and, as such, can be removed. Therefore, a fitness function for a given sequence  $S$  should follow a distribution proportional to the degree of optimality with respect to the average pattern.

Lets define a heuristic for calculating an estimation of the distance to the ground truth of a member  $s_i \in \nu \in S$ , given a matched period  $\bar{\nu} \in \mathcal{M}(\nu, \mathcal{P})$ . Hence,

$$\mathfrak{H}(\nu, i, \bar{\nu}) = \begin{cases} \min(|k - i \frac{|\bar{\nu}|}{|\nu|}|), & \text{if } \exists k / \nu(i) = \bar{\nu}(k) \\ \infty, & \text{otherwise} \end{cases}, \quad (6.21)$$

where  $||.||$  is the absolute value and  $|\vec{t}|$  is the size of the vector  $\vec{t}$ . Let

$$\mathfrak{F}(\nu, i, \mathcal{P}) = \sum_{\bar{\nu} \in \mathcal{M}(\nu, \mathcal{P})} \mathfrak{H}(\nu, i, \bar{\nu}) \quad (6.22)$$

be the *sequence member fitness* function, given a set of training patterns  $\mathcal{P}$ . The latter attempts to penalise the members on  $\nu$  that do not fit the patterns on  $\mathcal{M}(\nu, \mathcal{P})$ . Thus,  $\mathfrak{F}(\nu, i, \mathcal{P}, f_e)$  is the fitness of the  $i$ -th member of  $\nu$ . In order to calculate the fitness of the whole period, the *period fitness*

$$\mathfrak{F}(\nu, \mathcal{P}) = \sum_{\nu(i) \in \nu} \mathfrak{F}(\nu, i, \mathcal{P}) \quad (6.23)$$

is calculated for each  $\nu$  in a sequence  $S$ . That is,

$$\mathfrak{F}(S, \mathcal{P}) = \sum_{\nu \in \mathcal{N}(S)} (\mathfrak{F}(\nu, \mathcal{P})), \quad (6.24)$$

is the *sequence fitness* function that gives a final fitness value for a sequence  $S$ , given a series training patterns  $\mathcal{P}$ .

In the field of GAs, phenotypes or candidate solutions are normally encoded as sequences of *genes* with binary values, thus making the task of mutation and crossover simple. In the case that is being addressed, individuals are encoded as sequences of string elements.

### 6.2.3 A naive algorithm for extraneous member deletion on string sequences

While periodic sequence analysis and sequence fitness have already been defined, the question of what to do with *unfit* sequences remains open. Since the only source to account for fitness of a sequence is the presence of extraneous members within a periodic sub-sequence, it seems natural that the answer is to remove them. A trivial fashion to achieve this is defined in Algorithm 2. During each iteration, all members  $s_i$  marked as extraneous (i.e.,  $\mathfrak{F}(s_i, \nu, \mathcal{N}) > 0$ ) are removed from the sequence. The algorithm stops when the sequence does not contain any extraneous member ( $\mathfrak{F}(S, S^{Tr}, f_e) > 0$ ).

This approach has an important drawback because an extraneous sequence member to a period may in turn correspond to a different period within the same sequence. Take

**Algorithm 2** A naive algorithm for periodic sequence analysis

---

```

function NaivePSA( $S, \mathcal{P}$ )
   $f_n \leftarrow \mathfrak{F}(S, \mathcal{P})$  ▷ Calculate overall sequence fitness
   $f_{n+1} \leftarrow 0$ 
  while  $f_{n+1} \neq f_n$  do ▷ Iterate while fitness is different between consecutive sequences
     $\mathcal{N} \leftarrow \mathcal{N}(S)$ 
    for  $\nu \in \mathcal{N}$  do
       $i = \operatorname{argmax}_i \mathfrak{F}(\nu, i, \mathcal{P})$  ▷ Find the member with the largest fitness value and ...
       $S \leftarrow \{s_1 \dots s_{i-1}\} \cup \{s_{i+1} \dots s_n\}$  ▷ ... remove it from the original sequence
    end for
     $f_{n+1} \leftarrow \mathfrak{F}(S, \mathcal{P})$  ▷ Calculate fitness of the new sequence
  end while
  return  $S$ 
end function

```

---

$$S' = \{ABABABACD(A)CDCDC\} \quad (6.25)$$

as an example of a modification of the sequence in (6.18) by inserting the out-of-sequence extraneous member  $A$  between  $D$  and  $C$  in  $S'_1$ . Thus,

$$\mathcal{N}(S') = \{\{ABA\}\{ACDA\}\{CDAC\}\{CDC\}\}. \quad (6.26)$$

This sequence member, as seen in Section 6.1, is an extraneous one, as  $\mathcal{N}(S') \neq \mathcal{N}(S)$ . However, both new cycles  $\{ACDA\}$  and  $\{CDAC\}$  will match (according to  $\mathcal{M}$  in (6.20)) with the training cycles  $\{ABA\}$  and  $\{CDC\}$ , respectively, since the heading members are the same. That is,

$$\begin{aligned} \mathcal{M}(\{ACDA\}, \{\{ABA\}\{CDC\}\}) &= \{\{ABA\}\} \\ \mathcal{M}(\{CDAC\}, \{\{ABA\}\{CDC\}\}) &= \{\{CDC\}\}. \end{aligned} \quad (6.27)$$

This means that both  $C$  and  $D$  in  $\{ACDA\}$  and  $A$  in  $\{CDAC\}$  will all be identified as extraneous within their respective cycles and removed by Algorithm 2. The result will be

$$\text{NaivePSA}(S', \mathcal{P}) = \{ABABA[C][D](A)CDCDC\}, \quad (6.28)$$

which shows not only that the extraneous member ( $A$ ) has not been removed, but also  $[C]$  and  $[D]$  were incorrectly removed.

#### 6.2.4 An approach based on the Needleman-Wunsch algorithm for periodic sequences

To the best of the authors' knowledge, none of the approaches for exact or approximate string matching were designed for periodic sequences. The issue lies in the fact that single instances of each sub-sequence are matched against. It is herein proposed a modified version of the Needleman-Wunsch, shown in Algorithm 3.

The basic idea of `NeedlemanWunsch*` is to decompose the entry sequence  $S$  sequentially in candidate periods, i.e., periodic sub-strings headed by the same element as any  $s_i$  of the training patterns  $\mathcal{P}$ . Then, each candidate period is matched against  $s_i$  by the original Needleman-Wunsch algorithm and the results are concatenated. However, issues may arise when period heading members are missing in  $S$ .

### 6.3 A genetic algorithm for periodic sequences analysis

In this section, a genetic algorithm (GA) for extraneous member removal on periodic sequences is presented. It is presented in general terms, so its domain is defined on any periodic sequence  $S$  from a given alphabet  $\mathcal{A}$ . First, a naive algorithm for extraneous members removal is presented, in order to compare the proficiency of the proposed algorithm. Then, an introduction to GAs is outlined, involving the concepts of chromosome fitness, population, crossover, mutation and selection. Then, implementations of each of these concepts into the addressed problem are detailed. These include the definition of the fitness heuristic for candidate periodic sequences derived from the original.

#### 6.3.1 Genetic algorithms

GAs [31] are domain-independent, evolutionary search mechanisms inspired by the adaptive power of the concepts of *evolution*, *crossover*, *mutation* and *survival of the fittest* in genetics. Genetic algorithms are thus loosely based on the process of *natural selection* and conform a search

---

**Algorithm 3** A variant of the Needleman-Wunsch algorithm for periodic sequences

---

```

function NeedlemanWunsch*( $S, \mathcal{P}$ )
   $S_1 \leftarrow$ 
   $S_2 \leftarrow S$ 
  for all  $\nu \in \mathcal{P}$  do
     $S_1 \leftarrow \{S_1, \nu\}$ 
  end for
   $p_1 \leftarrow 1$ 
   $p_2 \leftarrow p_1 + 1$ 
  while  $S[p_2] \neq S[p_1]$  and  $p_2 \leq |S|$  do
     $p_2 \leftarrow p_2 + 1$ 
  end while
  while  $p_1 \leq |S|$  do
     $next \leftarrow p_1 + 1$ 
    for all  $s_i \in S^{Tr}$  do
      if  $s_i[1] = S[p_1]$  then
         $s_1, s_2 \leftarrow$  NeedlemanWunsch( $s_i, S[p_1 : p_2]$ )
        for  $j = p_1..p_2 - 1$  do
           $S_1 \leftarrow \{S_1[1 : j], s_1[j - p_1], S_1[j + 1]\}$ 
        end for
         $next \leftarrow p_2$ 
        break
      end if
    end for
     $p_1 \leftarrow next$ 
    if  $p_1 \geq 1$  and  $p_1 \leq |S|$  then
       $p_2 \leftarrow p_1 + 1$ 
      while  $S[p_2] \neq S[p_1]$  and  $p_2 \leq |S|$  do
         $p_2 \leftarrow p_2 + 1$ 
      end while
    end if
  end while
  return  $S_1, S_2$ ;
end function

```

---

heuristic which purpose is to find an optimal solution to the stated problem, in a bounded amount of time. The idea of the heuristic behind a classic GA is to maintain a *population* of *candidate solutions* (or *phenotypes*) that changes through time. The initial population will typically be made of estimations or naive solutions and will evolve after each generation. The measurement for optimality of an individual (and, thus, the population) is called the *fitness function* and its definition often accounts for most of the success or failure of a GA [78]. Furthermore, for evolution to take place, *genetic operators* like *selection*, *crossover* between phenotypes and *mutation* of individuals should also be defined.

Algorithm 4 shows a GA including the above genetic operators. Observe the iterative nature

of the algorithm, operating through *generations* (where  $P_N$  is the  $N$ -th generation) and the specification of a number of *termination criteria*. In this case, only the maximum number of iterations ( $N_g$ ) is considered. An initial population  $P$  is specified, as well as the `Fitness`, `Selection`, `Crossover` and `Mutate` operators. Mutation over the individuals are performed with a probability  $M_p$ .

---

**Algorithm 4** A classic implementation of a GA

---

```

function GA( $P, M_p, N_g$ )
   $f_2 \leftarrow \text{Fitness}(P)$ 
   $f_1 \leftarrow f_2$ 
   $N \leftarrow 1$ 
   $P_1 \leftarrow P$ 
  while  $N + 1 < N_g$  do
     $N \leftarrow N + 1$ 
     $f_N \leftarrow f_{N-1}$ 
     $P_N \leftarrow \text{Selection}(P_{N-1})$ 
     $P_N \leftarrow \text{Crossover}(P_N)$ 
    for  $p \in P_N$  do
      if  $\text{Random} < M_p$  then
         $p \leftarrow \text{Mutate}(p)$ 
      end if
    end for
     $f_N \leftarrow \text{Fitness}(P_N)$ 
  end while
  return  $\text{Fittest}(P_N)$ 
end function

```

---

For Algorithm 4, it is still necessary to define the `Fitness`, `Mutate`, `Selection` and `Crossover` operators. The first is essential in a good GA, as a fitness function correctly reflecting the aptitude of a member of the population is more likely to converge to an optimal set of solutions. `Selection` makes this effective, choosing, among the fittest competitors in the population, the surviving individuals. `Crossover` and `Mutate` contributes towards the variety of the population and helps avoiding an early end due to iteration over local minima. The following subsections summarise each of these operators and formalise a GA for periodic sequence analysis in its entirety. For the following, complete replacement of current generation has been considered for population update.

### 6.3.2 Fitness operators for sequences and sequence members

A starting point for the fitness function of a sequence is the sequence fitness function  $\mathfrak{F}$  defined in (6.24).  $\mathfrak{F}$  uses the sequence member fitness defined in (6.22) of each period members to calculate

the sequence fitness of a given sequence  $\vec{r}$ .

However, there are certain situations that  $\mathfrak{F}$  does not address directly, for example, the effect of an extraneous member over adjacent, non-extraneous members. Take the sequence

$$S' = \{ABABABACDACDCDC\} \quad (6.29)$$

again as an example. The period  $\{ACDA\}$ , with  $A$  acting as period header, exists in  $S'$ . The member  $A$  is extraneous in this case. Nevertheless, the members  $C$  and  $D$  will be penalised with  $f_e$  as they are also extraneous to the cycle  $\{ABA\}$ . The effect of this situation can be clearly seen in the sequence

$$S'' = \{ABABABACDCDCDCA\}. \quad (6.30)$$

In this case, all members in the sub-string  $\{\dots, CDCDCDC, \dots\}$  will be penalised as extraneous, following the rationale in Section 6.2.2. This is an extreme example where the extraneous member located at one of the edges of the sequence. In these situations, the extraneous members will never be penalised, since the only periods they belong to are the ones they head. This ultimately results in a misleading fitness function and wrongly evolved intermediate and final solutions.

In order to attempt minimising the effect of this situations, the fitness operator will take into account that the more out-of-sequence extraneous member within a sequence there are, the less likely they are of actually being extraneous. It is assumed that the sequence will only have a few extraneous members (if any). A large percentage of members of a period being penalised indicates a situation similar to, e.g., that of (6.30). Therefore, the fitness value has to be reverted, i.e., the fittest members in the period will become the weakest and vice-versa.

Algorithm 5 shows an algorithm for *chromosome fitness* (sequence member) calculation. If the period  $\nu$  within a given chromosome  $p_i$  has a rate of extraneous members bigger than a given value  $r_e$ , with  $0 \leq r_e \leq 1$ , the value of the fitness of each member (for the given period) is inverted from 0 to  $f_e$  and vice-versa.

Finally, the *phenotype fitness* operator for the calculation of the total fitness of a sequence, given each member's fitness calculated separately, is defined as

**Algorithm 5** A chromosome fitness operator for the periodic sequence analysis GA

---

```

function CFitness( $s_i, S, \mathcal{P}, f_e, r_e$ )
   $fitness \leftarrow 0$ 
  for all  $\nu \in \mathcal{N}(S)$  do                                ▷ Visit all sequence periods in which  $s_i$  is contained
    if  $\nu_i \in \nu$  then
       $f \leftarrow \mathfrak{F}(\nu, i, \mathcal{P}, f_e)$                     ▷ Calculate  $\nu_i$ 's fitness within the period
       $N_e \leftarrow 0$  ▷ Calculate the number of potentially extraneous members of the period
      for all  $\nu_i \in \nu$  do
        if  $\mathfrak{F}(\nu, i, \mathcal{P}, f_e) > 0$  then
           $N_e \leftarrow N_e + 1$ 
        end if
      end for
      ▷ If the extraneous member rate is bigger than  $r_e$ , invert the fitness value
      if ( $N_e > |\nu| \cdot r_e$  and  $f = 0$ ) or ( $N_e \leq |\nu| \cdot r_e$  and  $f \geq 0$ ) then
         $fitness \leftarrow fitness + f$ 
      end if
    end if
  end for
  return  $fitness$ 
end function

```

---

$$\text{Fitness}(S, \mathcal{P}, f_e, r_e) = \sum_{s_i \in S} \text{CFitness}(s_i, S, \mathcal{P}, f_e, r_e). \quad (6.31)$$

**6.3.3 Two mutation operators for sequences**

*Mutation*, within the field of GAs, is the equivalent of the natural mechanism for spontaneous change of chromosomes in genetic chains. The meaning, in terms of GAs, is that one of the gene values of a candidate solution may be altered at some point of the execution (normally, between the formation of two subsequent generations). In the case of sequences, it is suggested to implement this operator as the one resulting in the elimination of one of the members  $s_i \in S$  (chosen randomly) of the phenotype to be mutated, i.e.,

$$\text{MutateRandom}(S) = \{s_1 \cdots s_{r-1}\} \cup \{s_{r+1} \cdots s_{|S|}\} \quad (6.32)$$

with  $r = \text{Random}(1, |S|)$ . This of course does not guarantee that the chosen landmark to be removed is the best. However, the GA, through evaluation of the sequence fitness and the

Selection and Reproduction operations, aims to converge to a final optimised population  $P_N$ , with an outstanding member ( $\text{Fittest}(P_N)$ ), that will evolve from weakly-fitted phenotypes to those with a higher chance of survival.

Prior knowledge can be added to the mutation operator, since the sequence member fitness is a good indicator of a likely extraneous member. The sequence fitness function  $\mathfrak{F}$  defined in (6.24) uses the sequence member fitness defined in (6.22) of each cycle members as an intermediate result. Let the mutation operator make use of this information, by removing a random sequence cycle member among those whose fitness is not null (i.e., a cycle member likely to be extraneous). Therefore,

$$\text{MutatePrior}(S) = \{s_1 \cdots s_{p-1}\} \cup \{s_{p+1} \cdots s_{|S|}\}, \quad (6.33)$$

where  $p = \text{argmax}_i \text{CFitness}(s_i, S, \mathcal{P}, 1, r_e)$ , is a mutation function for a periodic sequence analysis GA with prior knowledge of sequence member fitness.

The `MutatePrior` function removes the sequence member with higher sequence member fitness greater than zero (i.e., does not appear within any of the training periods matched with its own sequence period), rather than any sequence member.

#### 6.3.4 A probabilistic tournament selection operator for sequences

The aim of the `Selection` function is to recreate the concept of *natural selection*. A *probabilistic tournament selection* algorithm is used, as seen in Algorithm 6.  $n$  candidates from generation  $P$  are selected for *reproduction*. They are randomly subdivided into  $k$ -sized subsets, each of which compete in the tournament. A *best candidate probability*  $BC_p$  parameter is specified, so that the best candidate for the tournament is selected with a probability  $BC_p$ , the second best candidate is selected with a probability  $BC_p(1 - BC_p)$ , the third best candidate is selected with a probability  $BC_p(1 - BC_p)^2$ , etc.

---

**Algorithm 6** A probabilistic tournament selection algorithm for the periodic sequence analysis GA

---

```

function ProbabilisticTournamentSelection( $S, n, k, BC_p$ )
     $sel \leftarrow \emptyset$ 
    for  $i = 1..n$  do
         $candidates \leftarrow \emptyset$ 
        for  $j = 1..k$  do
             $candidates \leftarrow \{candidates \cup P(\text{Random}(1, |P - candidates|))\}$ 
        end for
         $candidates \leftarrow \text{SortByFitness}(candidates)$ 
         $j \leftarrow 1$ 
        while  $\text{Random} > BC_p * (1 - BC_p)^j$  and  $j > 0$  do
             $j \leftarrow j + 1$ 
        end while
        if  $j > 0$  then
             $sel \leftarrow \{sel \cup candidates(j)\}$ 
             $S \leftarrow S - candidates(j)$ 
        end if
    end for
    return  $sel$ 
end function

```

---

### 6.3.5 A single point crossover operator for sequences

Finally, a *single point crossover* function has been used to create new candidates from existing ones. The aim of crossover is to emulate the natural mating of two members of a generation in order to produce new individuals. Single point crossover consists of selecting a crossover point inside each predecessor's chromosome chain and swap the two sub-chains beyond that point of either parent to the two children. Take the two sequences

$$S_1 = \{ABACCDAC\} \quad (6.34)$$

$$S_2 = \{ABCACDDC\} \quad (6.35)$$

as an example. If the crossover point is chosen as the position in the middle of the sequences, then the two sequences

$$S'_1 = \{ABACCDDC\} \quad (6.36)$$

$$S'_2 = \{ABCACDAC\} \quad (6.37)$$

will be produced after single point crossover. This genetic operator is shown in Algorithm 7, so that  $\text{Crossover} \equiv \text{SinglePointCrossover}$ .

---

**Algorithm 7** A single point crossover reproduction algorithm for the periodic sequence analysis GA

---

```

function SinglePointCrossover( $S$ )
   $S \leftarrow \text{SortByFitness}(S)$ 
   $S' \leftarrow \emptyset$ 
  while  $|S| > 2$  do
     $parent_1 \leftarrow S(\text{Random}(1, |S|))$ 
     $parent_2 \leftarrow S(\text{Random}(1, |S - parent_1|))$ 
     $child_1 \leftarrow parent_1$ 
     $child_2 \leftarrow parent_2$ 
    for  $j = 1 \dots |parent_1|$  do
      if  $j < |parent_1|/2$  and  $j \leq |parent_2|$  then
         $child_1(i) \leftarrow parent_2(i)$ 
      end if
      if  $j \geq |parent_2|/2$  and  $j \leq |parent_2|$  then
         $child_2(i) \leftarrow parent_1(i)$ 
      end if
    end for
     $S' \leftarrow \{S' child_1 child_2\}$ 
     $S \leftarrow S - \{parent_1 parent_2\}$ 
  end while
  return  $S'$ 
end function

```

---

## 6.4 Experimental results

In this section, the proposed GA is evaluated in two different scenarios. Specifically, synthetic periodic strings and sequences of key body poses are used to test the efficacy of the proposed approach.

Lets denote as  $S^G = \{s_1, s_2, \dots, s_N\}$  the ground truth of a given periodic sequence  $S$  with pattern  $\mathcal{P}$ . Let  $\bar{S}$  be the sample to be analysed. Finally, let  $\bar{S}^s$  be the output solution of GA.  $\bar{R} = \{r_1, r_2, \dots, r_N\}$ , where  $r_i \in \{True, False\}$ , denotes whether the ground truth member  $s_i \in S^G$  has been removed or not from  $\bar{S}$ . Similarly,  $r_i^s \in \bar{R}^s$  denotes whether the ground truth member  $s_i \in S^G$  has been removed or not from  $\bar{S}^s$ . Then, the true positive ( $tp$ ), true negative ( $tn$ ), false positive ( $fp$ ) and false negative ( $fn$ ) metrics are calculated as

$$\begin{aligned}
 tp(\bar{S}^s, S^G, \bar{R}, \bar{R}^s) &= |\{s_i \in S^G / r_i^s = False\}| \\
 tn(\bar{S}^s, S^G, \bar{R}, \bar{R}^s) &= |\{s_i \in S^G / r_i = True \text{ AND } r_i^s = True\}| \\
 fp(\bar{S}^s, S^G, \bar{R}, \bar{R}^s) &= |\{s_i \in \bar{S}^s / s_i \notin S^G\}| \\
 fn(\bar{S}^s, S^G, \bar{R}, \bar{R}^s) &= |\{s_i \in S^G / r_i = False \text{ AND } r_i^s = True\}|
 \end{aligned} \tag{6.38}$$

and then the precision ( $p$ ), accuracy ( $a$ ), true negative rate ( $tnr$ ), false discovery rate ( $fdr$ ) and false negative rate ( $fnr$ ) are calculated as

$$\begin{aligned}
 p &= tp / (tp + fp) \\
 a &= (tp + tn) / (tp + tn + fp + fn) \\
 tnr &= tn / (tn + fp) \\
 fdr &= fp / (tp + fp) \\
 fnr &= fn / (tp + fn).
 \end{aligned} \tag{6.39}$$

Consider, as an example, the ground truth pattern

$$\mathcal{P} = [PFGYVK]^*[WNSZTQJ]^*[RUCOHDAB]^*[EXMIL]^*,$$

where the implementation of

$$S^G = [PFGYVK]^5[WNSZTQJ]^6[RUCOHDAB]^6[EXMIL]^5$$

would be the ground truth, non-mutated sequence

$$S^G = PFGYVKPFGYVKPFGYVKPFGYVKPWNSZTQ$$

$$JWNSZTQJWNSZTQJWNSZTQJWNSZTQJWNSZTQ$$

$$JWRUCOHDABRUCOHDABRUCOHDABRUCOHDABR$$

$$UCOHDABRUCOHDABREXMILEXMILEXMILEXMI$$

$$LEXMILE.$$

An example of random mutation for the above would be

$$\bar{S} = PFGYV\{F\}K\{P\}F\cancel{G}YVK\{P\}\{Y\}PFG\{P\}YVKPF$$

$$GY\{P\}\cancel{V}KPW\cancel{N}SZT\cancel{Q}JWN\{S\}SZTQJ\{T\}WNSZTQ\cancel{J}\{J\}$$

$$WNSZTQJWN\{W\}SZTQJWNS\{S\}Z\{J\}TQJWR\{R\}UC$$

$$OHD\{B\}ABR\cancel{U}COHDAB\cancel{R}U\cancel{O}HDAB\{R\}RUCOHDABR$$

$$UCOHD\cancel{A}BR\{B\}\cancel{U}CO\{B\}H\{C\}DA\{O\}BREXMIL\{I\}EX$$

$$MIL\cancel{E}EXMIL\{L\}EXMI\cancel{L}EXMI\cancel{L}E,$$

where / denotes a deleted member and {·} denotes an extraneous inserted member, resulting in the testing sequence

$$\bar{S} = PFGYVFPPFYVPYPFGPYVKPFGYPPWSZTJW$$

$$NSSZTQJTWNSZTQJWNSZTQJWNWSZTQJWNSS$$

$$ZJTQJWRRUCOHDABABRCOHDABOHDABRRUCOH$$

$$DABRUCOHDABRBCOBHCDAOBREXMILIEXMILE$$

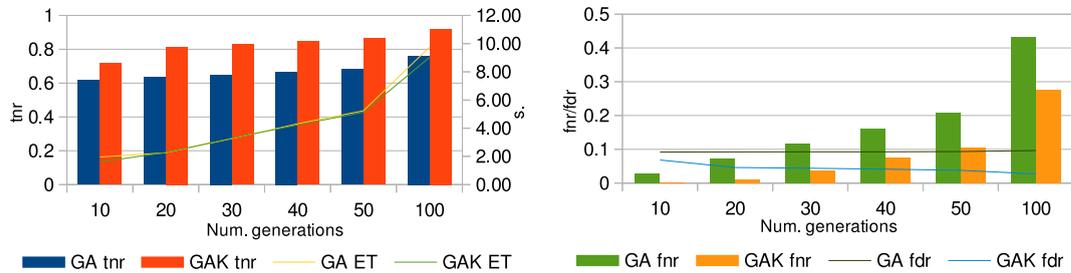
$$XMILLEXMIEXMIE.$$

#### 6.4.1 Evaluation on randomly generated periodic sequences

Experiments were performed on periodic string sequences with a random length. Specifically, 5 different training sequences were composed with the whole English alphabet, 4 to 10 number

of periods per periodic sub-sequences, 4 to 10 number of different periods and 4 to 10 items per period. Subsequently, 10 testing samples per training sequence were created and altered with in-period and out-of-period extraneous members to a rate of 0.1 and 0.2, respectively, and missing members to a rate of 0.1.

FIGURE 6.1: Charts showing performance of the GA and the  $GA^K$  for different number of generations, with a population size of 40 and a mutation probability of 0.7



(a) *tnr* and execution time (*ET*) for various number of generations

(b) *fnr* and *fdr* for various number of generations

The above experiments were performed by varying the value of a series of parameters of  $GA^K$ . Specifically, the effect of the number of generations, the population size, and the mutation probability were evaluated for a range of their values. Figure 6.1(a) shows a graphical comparison of the *tnr* and average execution time per testing sample (*ET*) achieved by GA and  $GA^K$  with different number of generations and a fixed population size of 40.

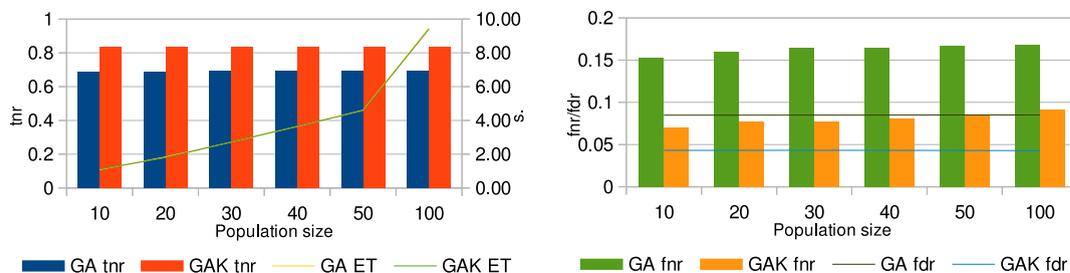
One of the observations from these results is that the number of generations affects (seemingly proportionally) the average execution time per sample, while specificity starts saturating with more than 40 generations. An acceptable performance can be achieved from this point.

Figure 6.1(b) shows *fnr* and *fdr* for the same experiments. It can be seen that the false negative rate becomes increasingly higher as a function of the number of generations for the GA, but is consistently very small (less than 1%) for the  $GA^K$ . Additionally, the false discovery rate shows a decreasingly dependent relation with the population size and is relatively small with 40 generations or less.

The same analysis was performed for a fixed number of generations (40) and a range of population sizes. Results can be seen in Figure 6.2.

These results are very interesting from the point of view of performance. In terms of *tnr*, a negligible variance can be observed among a wide range of population sizes, whereas the difference in execution time is large. This means that the working point for the  $GA^K$  can be

FIGURE 6.2: Chart showing performance of the GA and the  $GA^K$  for different population sizes, with 40 generations and a mutation probability of 0.7



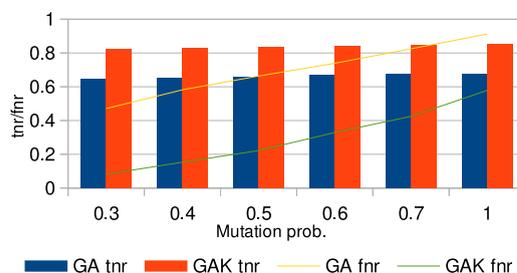
(a) *tnr* and execution time (*ET*) for various population sizes

(b) *fnr* and *fdr* for various population sizes

modified to a smaller population size to minimise the execution time, without incurring on a big generalisation error. Likewise, *fnr* and *fdr* show a similar consistency for both GA and  $GA^K$ .

Finally, experiments on a range of mutation probabilities around the random probability were also tested. GAs would normally have a probability of mutation of  $M_p < 0.5$  per iteration. Results can be seen in Figure 6.3.

FIGURE 6.3: Chart showing performance of the GA and the  $GA^K$  for different mutation probabilities, with 40 generations and 10 individuals per generation



It can be appreciated that better results, in terms of false negative rate, are achieved when alleles are mutated as seldom as possible. On the other hand, the *tnr* remains almost constant.

The *Needleman-Wunsch* algorithm [83] was used to benchmark the approach with the same training and testing samples, due to its ability of matching sub-sequences, rather than sub-strings, with a given pattern. However, it is known *a priori* that *Needleman-Wunsch* does not contemplate the repetition of sub-sequences within the look up text. Henceforth, the proposed periodic *Needleman-Wunsch* algorithm (*Needleman-Wunsch\**) was used as well. Additionally, a classic implementation of the *Smith-Waterman* algorithm [107] was also used to compare the efficiency of the algorithms.

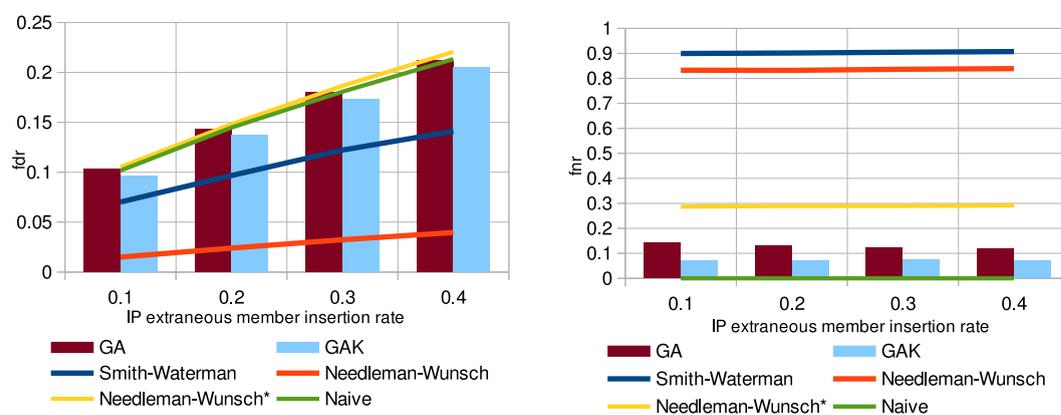
Table 6.1 shows the values for  $a$ ,  $r$ ,  $tnr$ ,  $fnr$  and  $fdr$ , as well as the average execution time per testing sample, achieved with the above mentioned algorithms and the Naive algorithm, the GA and the  $GA^K$  with a population size of 20 and 40 generations for random periodic sequences. *None* indicates the average metrics for the original testing samples before sequence analysis.

TABLE 6.1: Comparison of accuracy ( $a$ ), recall ( $r$ ), true negative rate ( $tnr$ ), false positive rate ( $fnr$ ), false discovery rate ( $fdr$ ) and average execution time ( $ET$ ) (in  $s$ ) achieved using several different methods for random, periodic sequence analysis. GAs run over 40 generations of 10 individuals each and a mutation probability of 0.3

Method	$a$	$r$	$tnr$	$fnr$	$fdr$	$ET$
None	0.90	1.00	0.50	0.00	0.11	-
Smith-Waterman	0.27	0.09	0.98	0.91	0.04	0.0013
Needleman-Wunsch	0.32	0.16	0.98	0.84	0.03	0.0019
Needleman-Wunsch*	0.58	0.54	0.76	0.46	0.10	0.0006
Naive	0.90	1.00	0.50	0.00	0.11	0.0016
GA	0.85	0.91	0.65	0.09	0.09	0.9202
$GA^K$	0.95	0.98	0.82	0.02	0.04	0.9311

Further trials with long periodic sequences with different rates of in-sequence extraneous members were conducted. Figure 6.4 shows the  $fdr$  and the  $fnr$  achieved by different algorithms on multiple experiment setups of random periodic sequences with different rate of in-period ( $IP$ ) extraneous members randomly inserted.

FIGURE 6.4: Chart showing a comparison between performance achieved by string matching algorithms and the GAs with periodic sequences with different rate of in-sequence extraneous members. GAs run in 50 generations, 10 individuals per generation and a mutation probability of 0.3



(a)  $fdr$  for varying in-sequence extraneous member insertion rates (b)  $fnr$  for varying in-sequence extraneous member insertion rates

It is clear that the level of obfuscation of the input sequences affects the performance of the GAs.

A higher value for  $fdr$  indicates that the approach leaves out more extraneous members. This situation can probably be improved with the use of more generations and/or a larger population size. However, the false negative rate shows to be almost constant and low for the GAs, as opposed to the other methods. This suggests that the proposed approach is very good identifying extraneous members and is not easily confused with proper members.

#### 6.4.2 Evaluation on periodic sequence analysis of key body pose sequences

Table 6.2 and Table 6.3 show precision ( $\pi$ ), accuracy ( $\alpha$ ), false positives ( $fp$ ) and false negatives ( $fn$ ) of identified key body poses achieved with the proposed approach for periodic sequence analysis.

TABLE 6.2: Comparison of true negative rate ( $tnr$ ), false positive rate ( $fpr$ ), false discovery rate ( $fdr$ ), false negative rate ( $fnr$ ) and average execution time per testing sample ( $ET$ ) (in  $s$ ) of identified key body poses achieved on periodic sequence analysis using the naive algorithm

Exercise	NaivePSA				
	$tnr$	$fpr$	$fdr$	$fnr$	$ET$
Ankles	0.89	0.10	0.32	0.46	0.0004
Arms	0.97	0.02	0.22	0.28	0.0002
Calves	0.94	0.05	0.26	0.40	0.0004
Inner thighs	0.98	0.01	0.16	0.26	0.0002
Shoulders	0.90	0.09	0.39	0.47	0.0002
<b>Average</b>	<b>0.94</b>	<b>0.09</b>	<b>0.27</b>	<b>0.37</b>	<b>0.0003</b>

TABLE 6.3: Comparison of true negative rate ( $tnr$ ), false positive rate ( $fpr$ ), false discovery rate ( $fdr$ ), false negative rate ( $fnr$ ) and average execution time per testing sample ( $ET$ ) (in  $s$ ) of identified key body poses achieved on periodic sequence analysis using both GAs with ( $GA^K$ ) and without (GA) prior knowledge

Exercise	GA					$GA^K$				
	$tnr$	$fpr$	$fdr$	$fnr$	$ET$	$tnr$	$fpr$	$fdr$	$fnr$	$ET$
Ankles	0.96	0.03	0.32	0.90	0.0166	0.95	0.04	0.30	0.82	0.0169
Arms	0.98	0.01	0.24	0.71	0.0065	0.98	0.01	0.23	0.59	0.0075
Calves	0.98	0.01	0.27	0.89	0.0169	0.97	0.02	0.29	0.81	0.0177
Inner thighs	0.98	0.01	0.17	0.70	0.0062	0.98	0.01	0.17	0.58	0.0073
Shoulders	0.96	0.03	0.38	0.82	0.0080	0.96	0.03	0.37	0.87	0.0085
<b>Average</b>	<b>0.97</b>	<b>0.02</b>	<b>0.28</b>	<b>0.80</b>	<b>0.0105</b>	<b>0.97</b>	<b>0.02</b>	<b>0.27</b>	<b>0.73</b>	<b>0.0116</b>

The experiments were tested against every result after 10-fold cross-validation for landmark identification and landmark refining, with the parameters that gave the best results for the latter

(see Section 5.6.2). Both naive and GA-based methods are compared. Results for two GAs are, in turn compared: one for a GA with no prior knowledge (i.e., using the `MutateRandom` mutation operator) and one with prior knowledge (i.e., using `MutatePrior`). The parameters chosen for the GA, tested empirically, were  $N_g = 10$  number of generations,  $BC_p = 0.65$  crossover probability,  $M_p = 0.30$  mutation probability and a population size  $|P| = 30$ .

Only results for the *Ankles* and *Calves* exercises show a small improvement, since there is no symmetry effect on *Arms*, *Inner thighs* and *Shoulders* (i.e., both sides of the body are exercised simultaneously). As expected, the naive algorithm incurs on a higher false negative rate, due to the issues explored in Section 6.2.3, although the false positives rate is lower (i.e., the GA fails to eliminate some out-of-sequence identified key body poses). The GA with prior knowledge show a very small improvement in terms of false positives and false negatives.

This effect can be more easily seen in Figure 6.5, where a graphical representation of this evaluation is shown. Specifically, identified key body poses ( $\bar{\chi}$ ) are compared with the result of a periodic sequence analysis using the naive algorithm ( $\bar{\chi}^S_N$ ), the GA with no prior knowledge ( $\bar{\chi}^S_G$ ) and the GA with prior knowledge ( $\bar{\chi}^S_{G^K}$ ).

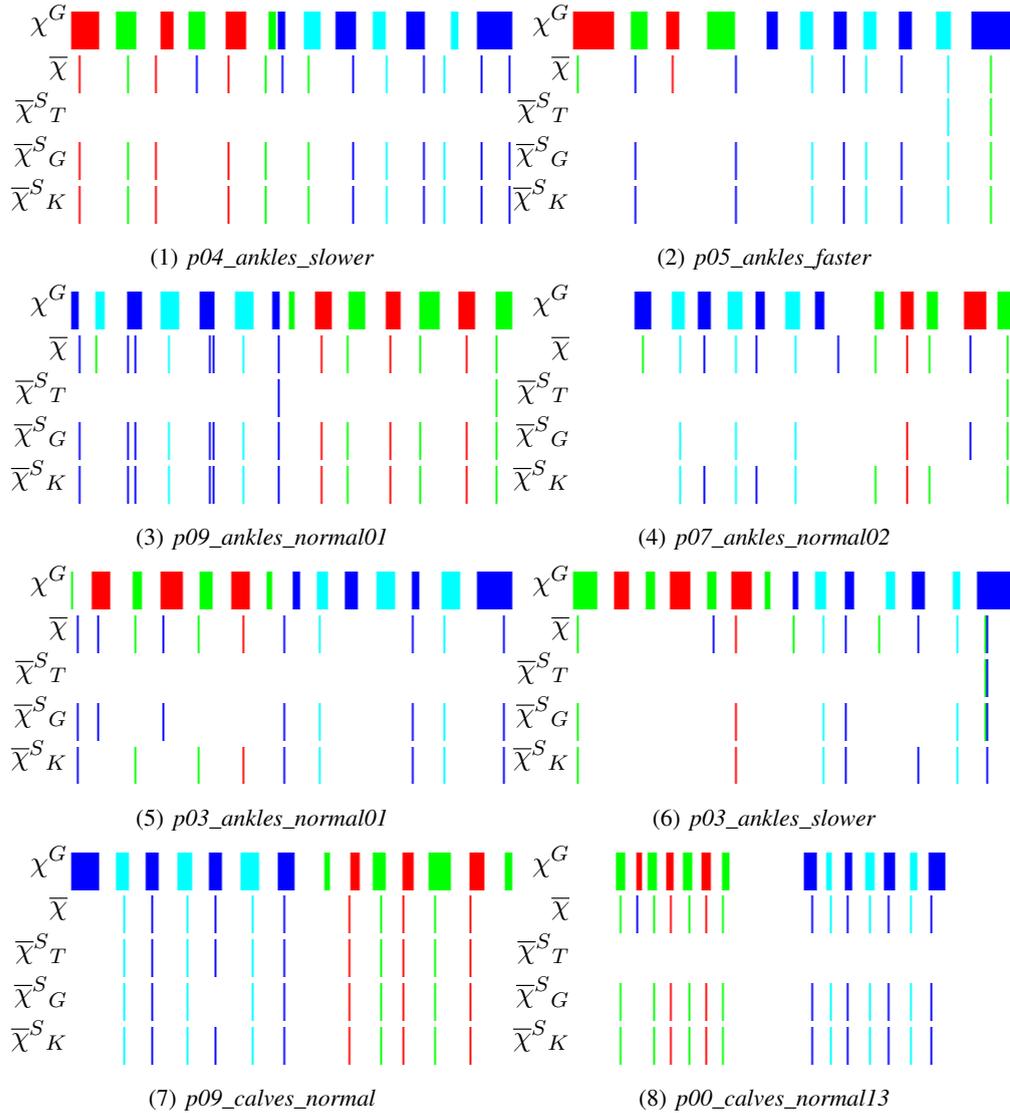
In most cases, the GAs succeed in removing extraneous elements. The main issue with the naive algorithm is that it removes correct members that are contained within sequence cycles headed by extraneous members (see Section 6.2.3). A notable failure of the GAs can be seen in Figure 6.5 (2) (with an otherwise better performance than the naive algorithm). Although the performances of GA and  $GA^K$  are very similar, Figure 6.5 (4) and Figure 6.5 (5) are notable examples of  $GA^K$  delivering better results than GA, effectively removing out-of-period extraneous elements.

## 6.5 Conclusions

In this chapter, an approach for periodic sequences analysis, based on a GA, has been presented. The problem of detecting and removing out-of-sequence members is specifically tackled. The approach is favourably benchmarked against classical string matching algorithms (where suitable sequences are transformed into strings).

Results drawn from experiments on sequences of key body poses are generally rather favourable due to the elementary nature of the problem (alphabets formed by 2-4 elements, samples with a

FIGURE 6.5: Comparison of results for periodic landmark sequence analysis between the trivial algorithm and the GA. Ground truth landmarks ( $\chi^G$ ), refined landmarks ( $\bar{\chi}$ ), periodic landmark sequence analysis using the trivial algorithm ( $\bar{\chi}^{S_T}$ ), a GA ( $\bar{\chi}^{S_G}$ ) and a GA with prior knowledge ( $\bar{\chi}^{S_K}$ ) represented with vertical markers of different colours, each of which representing a ground-truth, refined or identified landmark type, respectively



maximum length of 15 elements and an average false positive and negative rates of 2 and 3 per sample, respectively).

The approach, however, shows that there is room for improvement. Methods based on GAs are particularly inefficient because of the increased processing time, real-time applications being then unlikely. Therefore, improvement in the speed of convergence of the GA should be tackled. Furthermore, heavily obfuscated sequences may become unrecoverable, complicating the search of relevant data in real life for benchmarking. Therefore, in order to put the method into a clearer context and use real data, suitable applications should be addressed.

Proficiency achieved by  $GA^K$  over a series of experiments involving a number of parameters for the GA (population size, mutation probability and number of generations) has been evaluated and compared to classical string matching algorithms. Results suggest that optimal results may be achieved with a small population (of the order of 10) and number of generations (30). This reduces the processing time while producing comparable levels of specificity and sensitivity.

## Chapter 7

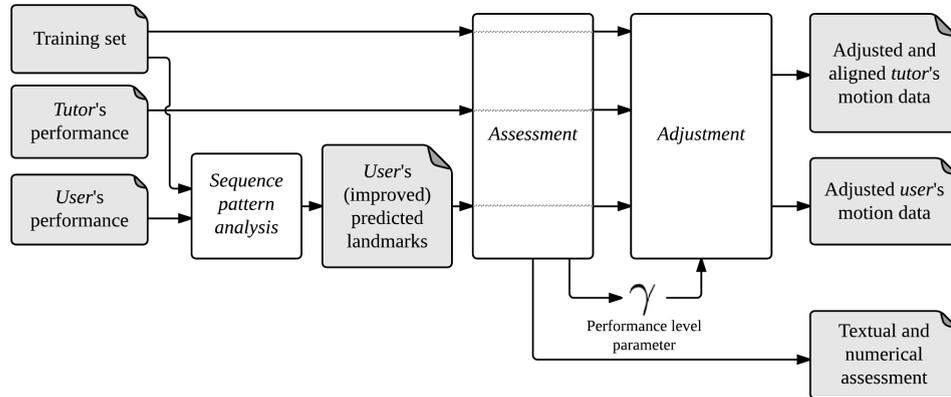
# Automatic, landmark-based feedback production based on adjusted parameters

This chapter presents an approach to automatic human motion feedback generation for basic stand-up exercises. A semi-supervised machine learning classifier is trained with ground truth, manually labelled sequences of key body poses. Before calculating feedback features to be learnt, motion synchronisation of previously identified landmarks (often incomplete and/or asymmetric) with the reference is done. Finally, an algorithm for motion adjustment of a performance to the tutor is proposed, in order to further enrich the feedback delivered to the user.

### 7.1 Overview of the methods for landmark assessment and motion adjustment

Figure 7.1 depicts the subsystems for periodic landmark sequence analysis, landmark assessment and motion adjustment. The mission of the periodic landmark sequence analysis stage is to improve the sequence of a user's identified landmarks produced in the previous stage (see Chapter 5) in order to remove *out-of-sequence* false positives, i.e., identified key body poses that do not match the cyclic pattern of an average landmark sequence of the motion class. These methods have been previously presented in Chapter 6.

FIGURE 7.1: System diagram depicting the periodic landmark sequence analysis, landmark assessment and motion adjustment stages



In Section 7.2, a method for automatic production of textual feedback (landmark assessment) for an improved landmark sequence is presented (*Assessment*). In this stage, an *assessment machine learning classifier* is trained with assessment features calculated from the difference between both user's and tutor's performances in order to infer a numerical assessment level for each involved body joint. These numerical values are then translated into natural language feedback and delivered to the user. A performance level parameter, as a way of summarising the current level of performance of the user, is calculated upon the assessment.

Finally, motion adjustment of the user's performance to that of the tutor is explained in Section 7.3 (*Adjustment*). In order to provide the user with visual feedback, an animation of their performance is shown along with a transformation to the tutor's key body poses. This will help the user to make comparison visually and to identify what needs to be improved. Additionally, a transformation of the tutor's performance to reflect user's level of performance is also shown as next execution's performance demonstration. Motion adjustment is calculated according to the value of the global performance level parameter ( $\gamma$ ) produced in the landmark assessment stage.

Results for these methods for landmark assessment, motion synchronisation and motion adjustment are described in Section 7.4, where different algorithms and parameters are evaluated and compared.

The foundations for this approach have been detailed in Chapter 5. As a reminder, a sequence of identified landmarks  $\bar{\chi}$  is produced by a semi-supervised machine learning classifier, being

$$\bar{\chi} = \{\chi_1 \dots \chi_N\}, \quad (7.1)$$

with

$$\chi_i = \{\text{FRAME}(t), \text{TYPE}(t)\} \forall \chi_i \in \bar{\chi} \quad (7.2)$$

and

$$\text{FRAME}(\chi_i) < \text{FRAME}(\chi_{i+1}) \forall \chi_i \in \bar{\chi}. \quad (7.3)$$

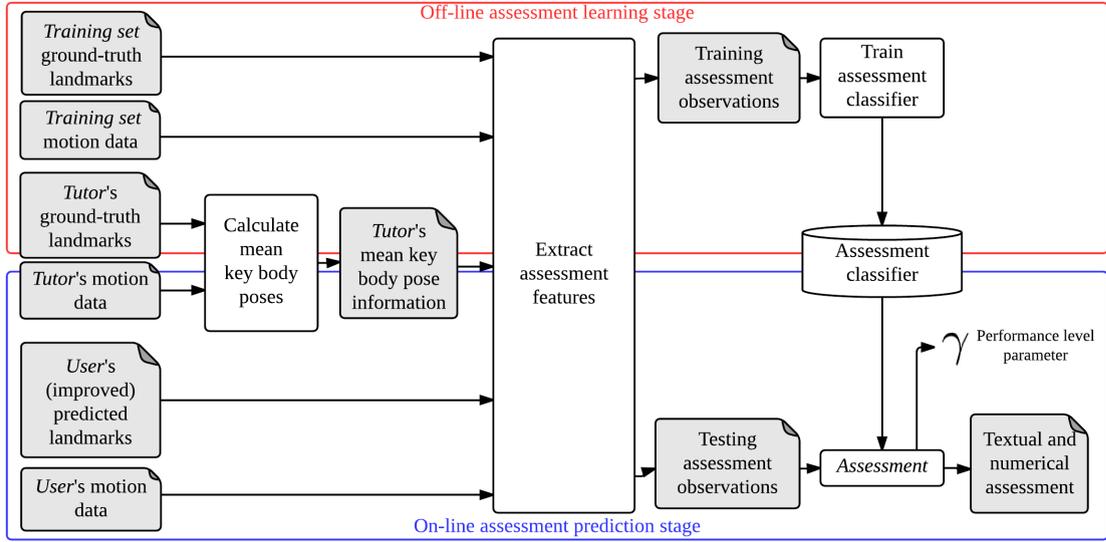
This gives a series of time frames of the user's motion data and a landmark type (i.e., range of motion (RoM) stretch limit) within the performance. This result simplifies enormously the task of retrieving specific key body poses from the input user performance for matching against those of the tutor.

## 7.2 Automatic production of textual assessment for synchronised motion series

In this section, a novel method for automatic production of numerical assessment of a given sequence of key body poses, with respect to an off-line trained motion data base, is explained. Assumptions for an incomplete, inaccurate set of identified landmarks ( $\bar{\chi}$ ) are made and prior motion synchronisation with the tutor of all training samples and testing samples constitutes a pre-requisite. The method, similar to that followed in Section 5.4 for candidate landmark feature learning and identification, is based on a multi-class, multi-target machine learning classifier based on extracted features of both tutor and user.

A diagram depicting how this stage works is shown in Figure 7.2. Before the assessment is done, the *Assessment classifier* is trained with training observations of *assessment features* per ground truth landmark extracted from the training samples. These assessment features are calculated upon comparison of the motion data with the tutor's *mean key body pose per landmark type*. Then, the testing observations are produced from the assessment features of the user's motion data and identified landmarks, following the same procedure. These testing observations are classified by the assessment classifier and numerical assessment is produced followed by textual feedback. Finally, the value for the performance level parameter is updated according to the calculated landmark assessment.

FIGURE 7.2: System diagram depicting the landmark assessment stage



### 7.2.1 Training observations and testing observations for assessment modelling

In order to automatically infer an assessment level, a similar method to that explained in Section 5.4.3 for training observations and testing observations is used. The training observations are extracted based on the difference between the values of the data source  $M^A$  (i.e., either the Euler angles  $E$  or the quaternions  $Q$ ) of the training samples and the tutor –  $\vec{M}^{ATr} = \{M_i^{ATr}\}$  and  $M_T^A$ , respectively. Likewise, the testing observations are extracted from the difference between the values of the data source of the testing sample ( $M^{ATs}$ ) and the tutor. The observations  $\mathcal{O}^A$  of a simple training sample or testing sample are then defined as

$$\begin{aligned} \mathcal{O}^A(M_P^A, M_T^A, T, AL_{\chi^G}, \chi_T^G) = \{ \{ t, \\ \{ \bar{P}(M_T^A, \chi_T^G, \text{TYPE}_{\chi^G}(t))(j_i) - M_P^A(t, j_i) \}, \\ \text{TYPE}_{\chi^G}(t), \\ \{ AL_{\chi^G}(t, j_i) \} \\ \} \} \forall t \in \mathbb{T}, j_i \in \mathbb{J}, \end{aligned} \quad (7.4)$$

where  $\bar{P}(M_T^A, \chi_T^G, \tau)(j_i)$  is the mean value of all tutor's poses in its ground truth landmarks of landmark type  $\tau$  in the  $i$ -th body joint (see (5.30) on Section 5.5.2). In this case, the observation class vector  $\vec{y} \in Y$  of each observation in  $\mathcal{O}_A$  belongs to the domain of  $Y_{\mathbb{J}} = \{y_1 \dots y_{N_{\mathbb{J}}}\}$ , with

$y_i \in \{none, 1 .. N_A\}$ . That is, an assessment level attribute per involved body joint  $j$ . This way, the  $AL_{\chi^G}$  function is defined as the landmark assessment descriptor, being

$$AL_{\chi^G}^{\mathbb{T}r}(t, j_i) = \begin{cases} AL(\chi^G_i, j_i), & \text{if } \exists \chi^G_i \in \chi^G / \text{FRAME}(\chi^G_i) = t \\ none, & \text{otherwise} \end{cases} \quad (7.5)$$

in the case of the training observations, and

$$AL_{\chi^G}^{\mathbb{T}s}(t, j_i) = none, \quad (7.6)$$

in the case of the testing observations, with  $AL_{\chi^G}(t, j_i) \in \{none, 1, ..N_A\}$ .

In other words, AL is an assessment level value assigned in accordance with the pose of the body joint  $j_i$  in the frame  $t$ . Given a series of values  $M^{\vec{A}\mathbb{T}r}$  of the training samples's motion source  $M_i^{A\mathbb{T}r} \in M^{\vec{A}\mathbb{T}r}$ , their ground truth landmarks  $\chi^{\vec{G}} = \{\chi^G_i\}$ , their selected training frames  $\vec{\mathbb{T}r} = \{\mathbb{T}r_i\}$  and the angles of the tutor  $M_T^A$ , the training observations for landmark assessment are chosen as

$$\mathcal{O}^{A\mathbb{T}r} = \{\mathcal{O}^A(M_i^{A\mathbb{T}r}, M_T^A, \mathbb{T}r_i, AL_{\chi^G_i}^{\mathbb{T}r}, \chi^G_i)\} \forall i. \quad (7.7)$$

Equivalently, given the Euler angles  $E^{\mathbb{T}s}$  of the testing sample, its ground truth landmarks  $\chi^G_{\mathbb{T}s}$  and its identified landmarks  $\bar{\chi}$ , the testing observations are chosen as

$$\mathcal{O}^{A\mathbb{T}s} = \mathcal{O}^A(M^{A\mathbb{T}s}, M_T^A, \text{FRAME}(\bar{\chi}), AL_{\chi^G_{\mathbb{T}s}}^{\mathbb{T}s}, \chi^G_{\mathbb{T}s}). \quad (7.8)$$

### 7.2.2 Assessment of identified landmarks

The assessment machine learning classifier has to deal, this time, with a number of observation classes, rather than just one, as was the case of the landmark identification machine learning classifier. Let

$$\Psi_i(\{x_{ij}^{\mathbb{T}r}\}, \{y_i^{\mathbb{T}r} j\}, x_i^{\mathbb{T}s}) = y_i^{\mathbb{T}s} \forall j = 1..N_{\mathbb{T}r}, \quad (7.9)$$

with  $\{y_i^{\mathbb{T}^s} \forall i\} \in \vec{Y}$ , be the definition of a *sub-machine learning classifier*  $\Psi_i$  that produces an assessment level  $y_i^{\mathbb{T}^s}$  as a function of the  $i$ -th training feature and the assessment observation class label  $y_i$ . Then, let

$$\Psi(\vec{x}^{\mathbb{T}^r}, \vec{Y}^{\mathbb{T}^r}, x^{\mathbb{T}^s}) = \{\Psi_i(\{x_{ij}^{\mathbb{T}^r}\}, \{y_i^{\mathbb{T}^r} j\}, x_i^{\mathbb{T}^s}) \forall j = 1..N_{\mathbb{T}^r}, i = 1..n, \quad (7.10)$$

be the landmark assessment machine learning classifier  $\Psi$  as a function of the  $n$  feature values of the  $N_{\mathbb{T}^r}$  training observations  $x^{\mathbb{T}^r}$  and the  $n$  feature values of the testing observations  $x^{\mathbb{T}^s}$ .

Let  $\mathbb{A}(\mathcal{O}^{A\mathbb{T}^r}, o) = y$ , with  $y \in Y$ , denote the output of a multi-class, multi-target machine learning classifier  $\mathbb{A}$  –trained with training observations  $\mathcal{O}^{\mathbb{T}^r}$ – over an unseen testing observation  $o$ , so that

$$\mathbb{A}(\mathcal{O}^{A\mathbb{T}^r}, o) = \Psi(\{\{o^{\mathbb{T}^r}(i)\}\}, \{\{\mathbb{AL}(o^{\mathbb{T}^r}(i), j_i)\}\}, \{o(i)\}) \forall o^{\mathbb{T}^r} \in \mathcal{O}^{A\mathbb{T}^r} \forall i = 2..n \forall j_i \in \mathbb{J}. \quad (7.11)$$

Let

$$\bar{\chi} = \{\{\text{FRAME}(t), \text{TYPE}(t), \{\mathbb{AL}(t, j_i)\}\}\} \forall t \in \mathbb{T}^s \forall j_i \in \mathbb{J} \quad (7.12)$$

be the assessed landmarks classified by the assessment machine learning classifier. It is defined as

$$\bar{\chi}(\mathcal{O}^{A\mathbb{T}^r}, \mathcal{O}^{A\mathbb{T}^s}) = \{\{\text{FRAME}(o) = o(1), \text{TYPE}(o), \{\mathbb{AL}(o) = \mathbb{A}(\mathcal{O}^{\mathbb{T}^r}, o)\}\}\} \forall o \in \mathcal{O}^{\mathbb{T}^s}. \quad (7.13)$$

The classifier evaluated for this stage is the C4.5 classifier with AdaBoost (see Section 5.4.4).

### 7.2.3 A parameter to model the current overall level of performance

In order to provide the framework of an adaptive mechanism, the level of demand, or performance level parameter  $\gamma_{j_i}$ , for each body joint  $j_i$  needs to be adjusted to the current state of fitness

of the individual. This needs to be a function of the previous value of  $\gamma_{j_i}$  and the current and previous assessment levels of  $j_i$ . Therefore,

$$\gamma_{j_i}^n = \Gamma(\gamma_{j_i}^{n-1}, \{\text{AL}(\chi_i \in \bar{\chi}^n, j_i)\}) \forall j_i \in \mathbf{J}, \quad (7.14)$$

where  $n$  is the ordinal number of performance of the same individual through time.

#### 7.2.4 Feedback generation from landmark assessment: an assessment *meta-grammar* and an assessment tree

In order to easily convey understandable feedback to the user, the numerical assessment levels inferred in  $\bar{\chi}$  need to be translated into meaningful words. Furthermore, the comments will depend not only on the value of each  $\text{AL}(\chi_i, j_i) \forall \chi_i \in \bar{\chi} \forall j_i \text{ in } \mathbf{J}$ , but also on the landmark type of the identified landmark  $\chi_i$ . For instance, an assessment level of 3 on the pose of the `LEFT_KNEE` in a identified landmark, of landmark type 2, in the *Ankle stretches* exercise has a different meaning than an assessment level 3 on the pose of the `LEFT_KNEE` in a identified landmark, of landmark type 4, in the same motion class (in this case, they belong to different stages of the stretches).

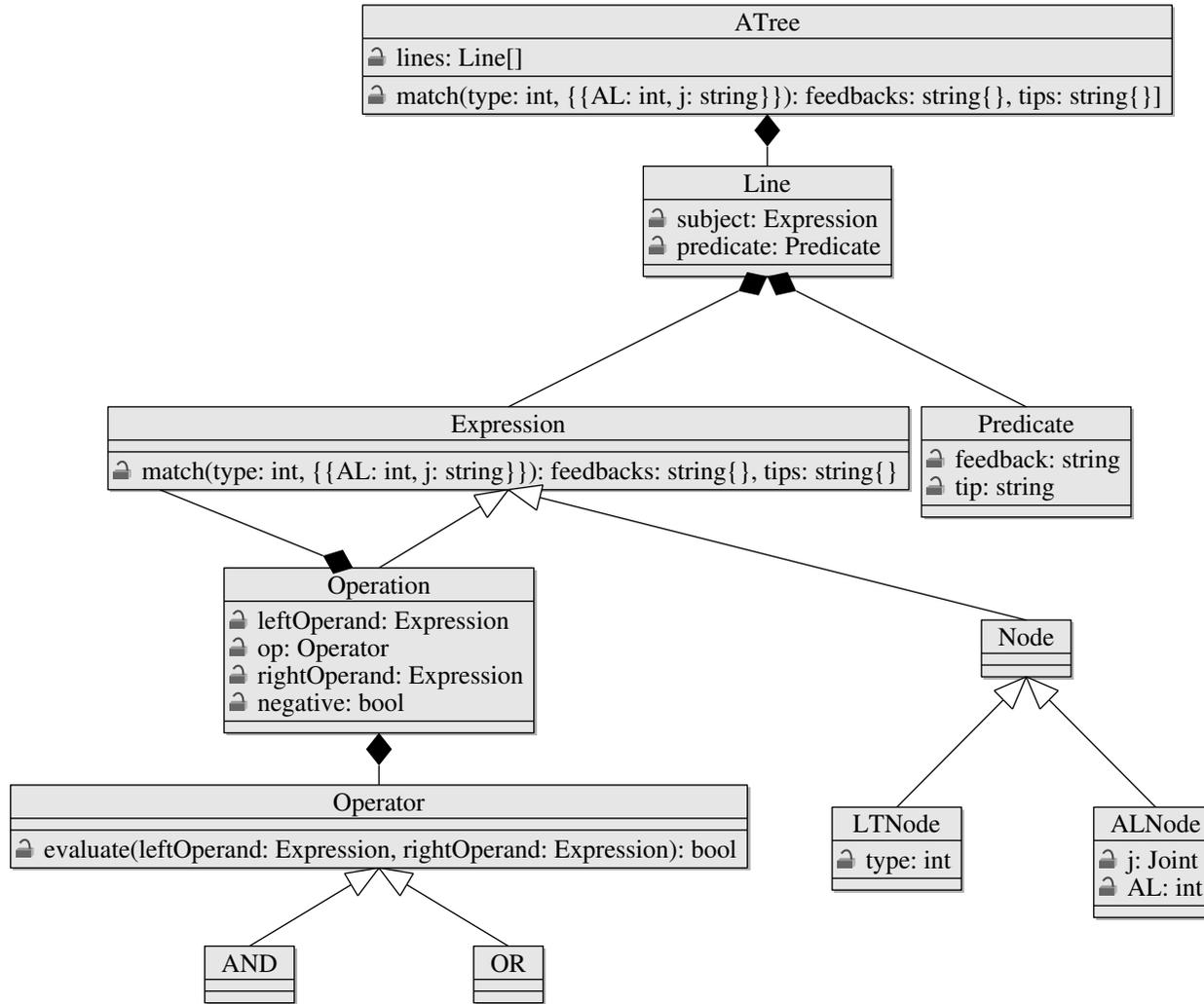
An expert system dedicated to analysing the above and infer a feedback *comment* and *tip* for each tuple  $\bar{\chi}_i = (\tau = \text{TYPE}(\chi_i), \{\{j_i, \text{AL}(\chi_i, j_i)\}\})$  for each exercise has been developed in the form of a grammar, defined by the context-free *meta-grammar* in Listing 7.2. The meta-grammar is used to parse an assessment grammar  $G$  and produce a hierarchised *assessment tree*, which in turn tries to match the conditions in  $\bar{\chi}_i$  and output a series of *comments* and *tips*. The latter are uttered to the user as a previous step to the evaluation demonstration and will be accompanied with synchronised visual feedback. To increase the level of naturalness of the method, tips are only uttered once. The grammars used to create the assessment tree for each exercise can be found in Appendix E.

LISTING 7.2: Back-Naur representation of the assessment meta-grammar

$$\begin{aligned}
\langle \text{line} \rangle & \models \langle \text{exp} \rangle = \langle \text{predicate} \rangle \\
\langle \text{exp} \rangle & \models ( \langle \text{exp} \rangle ) \mid ! \langle \text{exp} \rangle \mid \langle \text{exp} \rangle \langle \text{LOGICAL} \rangle \langle \text{exp} \rangle \mid \langle \text{node} \rangle \\
\langle \text{node} \rangle & \models \langle \text{landmarktype} \rangle \mid \langle \text{jointassessment} \rangle \\
\langle \text{landmarktype} \rangle & \models \text{type} \$ \langle \text{NUM} \rangle \\
\langle \text{jointassessment} \rangle & \models @ \langle \text{ID} \rangle \$ \langle \text{NUM} \rangle \\
\langle \text{predicate} \rangle & \models \langle \text{feedback} \rangle , \langle \text{tip} \rangle \mid \langle \text{tip} \rangle , \langle \text{feedback} \rangle \mid \langle \text{tip} \rangle \mid \langle \text{feedback} \rangle \\
\langle \text{feedback} \rangle & \models \text{F:} \langle \text{quotedliteral} \rangle \\
\langle \text{tip} \rangle & \models \text{T:} \langle \text{quotedliteral} \rangle \\
\langle \text{quotedliteral} \rangle & \models \text{``} \langle \text{STRING} \rangle \text{"} \\
\langle \text{LOGICAL} \rangle & \models \& \mid \mid \\
\langle \text{STRING} \rangle & \models \langle \text{CHAR} \rangle \langle \text{STRING} \rangle \mid \langle \text{NUM} \rangle \langle \text{STRING} \rangle \mid \text{' } \langle \text{STRING} \rangle \mid \lambda \\
\langle \text{ID} \rangle & \models \langle \text{CHAR} \rangle \langle \text{ID} \rangle \mid \lambda \\
\langle \text{CHAR} \rangle & \models \text{a} \dots \text{z} \mid \text{A} \dots \text{Z} \\
\langle \text{NUM} \rangle & \models \text{0} \dots \text{9} \text{ NUM} \mid \lambda
\end{aligned}$$

In order for an assessment tree to parse the entry assessment and yield fitness feedback in words, a class hierarchy for each parsed element of the parsed assessment grammar has been designed and is shown in Figure 7.3. Each class corresponds with an element in the specified grammar (*Expression*, *Operation*, *Node*, ...). The ANTLR library for Java [88] has been used to implement and parse the assessment grammars, as it provides the functionality of transforming each production rule into an instance of an object in parallel with the parsing process.

FIGURE 7.3: Class diagram for the assessment tree



Given a single landmark type  $\tau$  identified in the landmark identification process at a given time frame  $t$  and the corresponding assessment levels  $\{AL(j)\}$  produced in the landmark assessment stage, each *Line* in the assessment tree's top node *ATree* is visited and both *feedback* and *tip* are produced, or none. Whether a particular *Line* matches the  $\{\tau, \{AL(j)\}\}$  conditions is retrieved by invoking the *match()* method of the *Line*'s *subject*, which is an instance of an *Expression*. If it matches, then the feedback and tip in the *Line*'s *subject* are added to the text-to-speech output. Algorithm 8 shows the implementation of this method.

---

**Algorithm 8** Algorithm for expression matching with predicted conditions
 

---

```

function ExpressionMatch(exp,  $\tau$ ,  $\{\{AL(j)\}\}$ )
  if exp is a Operation then  $\triangleright$  If exp is an operation, evaluate the operands and the operation
    leftOp  $\leftarrow$  ExpressionMatch(exp.leftOperand)
    rightOp  $\leftarrow$  ExpressionMatch(exp.rightOperand)
    return exp.negative  $\oplus$  exp.op.evaluate(leftOp, rightOp)
  else
    if exp is a Node then  $\triangleright$  If exp is a node, try to match it against the conditions
      if exp is a LTNODE then return exp.type =  $\tau$ 
      else
        if exp is a ALNode then return  $AL(exp.j) = exp.AL$ 
        end if
      end if
    end if
  return false
end function

```

---

Algorithm 9 shows how the *Expressions* in each *Line*'s *subject* is matched with the entry conditions and both feedback and tips are produced.

---

**Algorithm 9** Algorithm for assessment tree parsing and matching with predicted conditions
 

---

```

function ATreeMatch(T,  $\tau$ ,  $\{\{AL(j)\}\}$ )  $\triangleright$  Initialise to empty feedback and no tips
  feedbacks  $\leftarrow \emptyset$ 
  tips  $\leftarrow \emptyset$ 
  for line  $\in T$  do  $\triangleright$  Try to match every line against the conditions
    if ExpressionMatch(line.subject,  $\tau$ ,  $\{\{AL(j)\}\}$ ) then  $\triangleright$  If the line matches,
    then add the associated feedback message and tip
      feedbacks  $\leftarrow \{feedbacks, line.predicate.feedback\}$ 
      tips  $\leftarrow \{tips, line.predicate.tip\}$ 
    end if
  end for
  return feedbacks, tips
end function

```

---

Finally, Algorithm 10 shows a general algorithm for the replay of the user's performance, given the user's skeletal data  $P$ , the exercise's assessment grammar  $G$  and a series of identified and assessed landmarks  $\bar{\chi}$ .

---

**Algorithm 10** Algorithm for on-line feedback generation
 

---

```

function PlayFeedback( $P, G, \bar{\chi} = \{t, \tau, \{\{AL(j)\}\}\}$ )
   $T \leftarrow \text{ANTLRParse}(G)$  ▷ Parse the assessment grammar
   $frame \leftarrow 0$ 
   $utteredTips \leftarrow \emptyset$ 
   $previousLandmark \leftarrow frame$ 
  while  $currentFrame < |P|$  do ▷ Show skeletal representation, frame by frame
    PaintSkeleton( $P, frame$ )
    if  $frame \in \bar{\chi}(t)$  then ▷ If there is a landmark ...
       $feedback, tips \leftarrow \text{ATreeMatch}(T, \bar{\chi}(frame))$  ▷ ... retrieve feedback, ...
      if  $feedback \neq \emptyset$  or  $tips \neq \emptyset$  then
        TextToSpeech( $feedback$ ) ▷ ... and utter it (if available)
        if  $tip \notin utteredTips$  then
          TextToSpeech( $tip$ ) ▷ Only utter each tip once
           $utteredTips \leftarrow \{utteredTips, tip\}$ 
        end if ▷ Ask the user whether replaying is necessary
        if InputMessage("Do you want me to repeat?") then
           $frame \leftarrow previousLandmark$ 
        else
           $frame \leftarrow frame + 1$ 
        end if
      end if
       $previousLandmark \leftarrow frame$ 
    else
       $frame \leftarrow frame + 1$ 
    end if
  end while
end function

```

---

Take the excerpt of the assessment grammar for the *Arm raises* exercise show in Listing 7.3 as an example. The ANTLR grammar parser will produce the assessment tree shown in Figure 7.4 and Figure 7.5.

---

 LISTING 7.3: Excerpt of the assessment grammar for *Arm raises*


---

```

type$2 & @LEFT_SHOULDER$2 ->F:"Your left arm should be raised a little bit higher up"
type$2 & @LEFT_SHOULDER$3 ->F:"Your left arm should be raised much higher up"
type$2 & @LEFT_SHOULDER$4 ->F:"You are not rising your left arm"
type$2 & @RIGHT_SHOULDER$4 ->F:"You are not rising your right arm"

type$2 & (@LEFT_SHOULDER$2 | @LEFT_SHOULDER$3 | @LEFT_SHOULDER$4 | @RIGHT_SHOULDER$2 |
  @RIGHT_SHOULDER$3 | @RIGHT_SHOULDER$4) ->T:"Look at the instructor and try to
  reach up with your arms on top of your head"

```

---

FIGURE 7.4: Assessment tree produced after ANTLR parsing of Listing 7.3

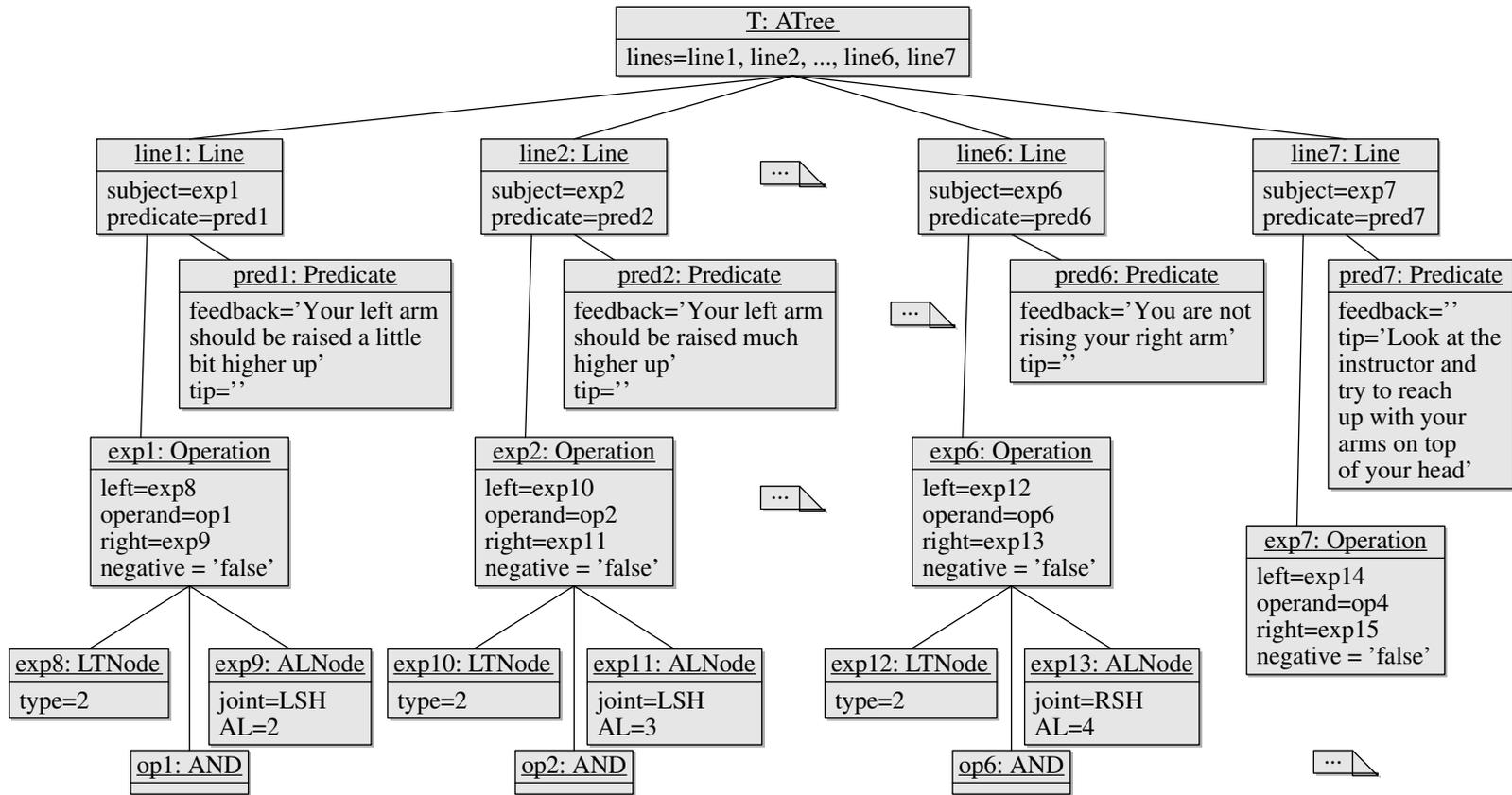
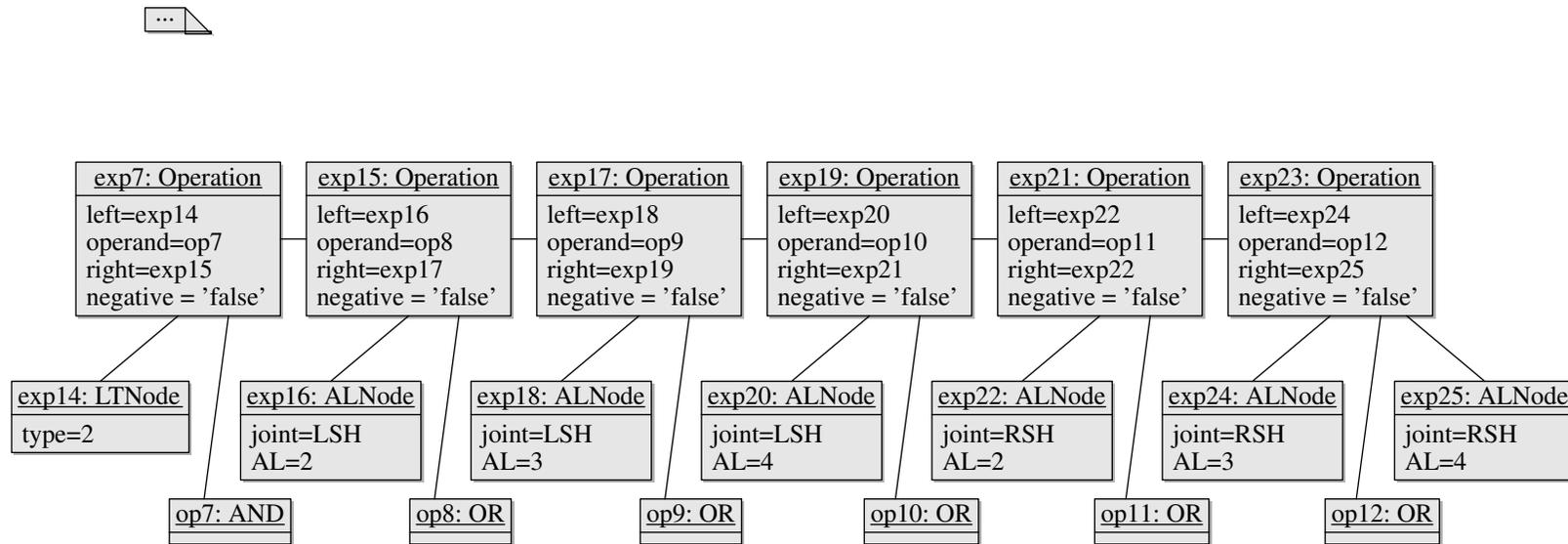


FIGURE 7.5: Assessment tree produced after ANTLR parsing of Listing 7.3 - *exp7* (continued from Figure 7.4)

### 7.3 Motion adjustment of natural demonstration for motivational research

In this section, a series of methods for motion synchronisation and motion adjustment are presented. The objective is to convey to the user performing physical fitness exercises visual information of the motion in the form of their animated skeletal appearance. Prior motion synchronisation of the tutor to the user is needed. Motion adjustment is conceived as a means to reconstruct the movement of the user with a modified appearance in the identified landmarks. The aim is to modify the reference movement (i.e., the tutor's) in order to reduce the differences between them.

FIGURE 7.6: System diagram depicting tutor's motion data motion synchronisation and motion adjustment stages

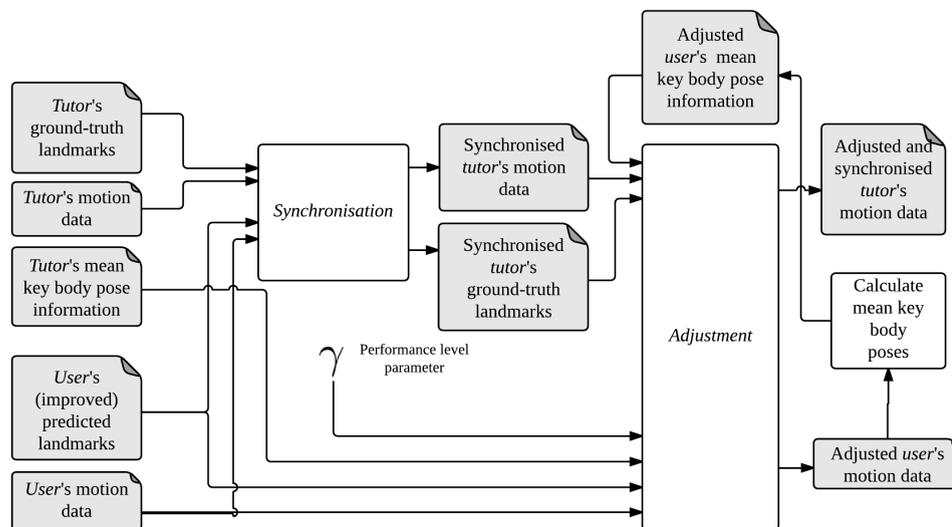


Figure 7.6 shows a depiction of the motion synchronisation (*Synchronisation*) and motion adjustment (*Adjustment*) stages. Motion synchronisation is performed on the basis of a likely incomplete sequence of identified landmarks and the tutor's ground truth landmarks are available. This pair of landmark sequences are aligned as a prior step to motion adjustment.

The *demonstration animation* consists of four super-imposed skeletal representations:

- The performance of the user,
- the performance of the user adjusted to the mean key body poses of the tutor's performance according to the value of  $\gamma$ ,
- the performance of the tutor and

- d) the performance of the tutor synchronised to that of the user and adjusted according to the value of  $\gamma$ .

With the animation, the user can instinctively compare the current performance to a reference, in order to improve their overall achievement level. The advantage of comparing *a*) against *b*) is that differences between both user's and tutor's appearances and orientations are removed, whereas visual matching between *a*) and *c*) is not as straight-forward. If a direct comparison with the tutor is desired, *c* will still provide a tailored demonstration. Motion adjustment is done to tailor the demand to an attainable level to avoid discouraging the user.

### 7.3.1 An algorithm to synchronise two incomplete and asymmetric landmark sequences

Given two landmark sequences  $\chi_1$  and  $\chi_2$ , a motion synchronisation between the two is defined as

$$\mathbb{S} = \{\mathbf{s}_1 \dots \mathbf{s}_M\}, \quad (7.15)$$

with  $\mathbf{s}_i = \{\{\chi_j, \chi_k\}\} \forall \mathbf{s}_i \in \mathbb{S}$ , where every pair  $\{\chi_j \in \chi_1, \chi_k \in \chi_2\}$  is a one-to-one matching between the elements of  $\chi_1$  and those of  $\chi_2$ .

A sequential approach has been followed in order to try overcome the constraints identified in (5.12) and (5.13). First, a method (`SynchroniseLandmarkSeries`) for synchronising two sequences of landmarks  $\chi_1$  and  $\chi_2$  as shown in Algorithm 11.

### 7.3.2 A method to synchronise two motion sequences, given their landmark sequences

In order to provide the user with visual feedback, the tutor's performance motion data ( $M_T$ ) should be synchronised with that of the user ( $M_P$ ). Let  $s_i$  be the motion information of a given motion data series in the time frame  $i$ . That is,  $s_i \in \mathbb{R}^D$ , with  $D$  being the number of relevant joints and degrees of freedom. Then, a motion data series  $M$  is defined as

$$M = \{s_1, \dots, s_F\}, \quad (7.16)$$

**Algorithm 11** Landmark sequence synchronisation algorithm

---

```

function SynchroniseLandmarkSeries( $\chi_1, \chi_2, N_1, N_2$ )
   $\mathbb{S} \leftarrow \emptyset$  ▷ Empty synchronisation sequence to begin with
   $lastAssigned \leftarrow \chi_1(N_1)$  ▷  $\chi_1(1)$  will be tried to be matched first
   $assigned \leftarrow Repeat(False, N_1)$  ▷ None has been matched to begin with
  for all  $\chi_j \in \chi_2$  do ▷ Iterate over the landmarks in  $\chi_2$ 
     $type_2 \leftarrow TYPE(\chi_j)$  ▷ Check, in order, all unassigned landmarks of  $\overline{\chi_1}$ 
    for all  $\chi_i \in \chi_1((lastAssigned \dots N_1] \cup [1 \dots lastAssigned))$  do
       $type_1 \leftarrow TYPE(\chi_i)$  ▷ Match if the TYPE is the same
      if  $type_1 = type_2$  and  $assigned(i) = False$  then
         $\mathbb{S} \leftarrow \{\mathbb{S}, \{FRAME(\chi_i), FRAME(\chi_j)\}\}$ 
         $assigned(i) \leftarrow True$ 
         $lastAssigned \leftarrow \chi_i$ 
        break
      end if
    end for
  end for
  return  $\mathbb{S}$ 
end function

```

---

i.e. a series of  $F$  of the instances in  $s_i$ . A dynamic approach to synchronise the motion data series  $M_T$  and  $M_P$  of a tutor and a user, respectively, of the same motion class is shown in Algorithm 12.

SynchroniseSubSequence is a function in the domain of  $\mathbb{R}^n$  which, given two motion sub-sequences  $S_1$  and  $S_2$ , probably of different lengths, fits the size of  $S_1$  to that of  $S_2$ , so that  $|S_1| = |S_2|$ , preserving as much temporal information as possible. This function is defined as

$$SynchroniseSubSequence(S_1, S_2) = \{S_1(\frac{i \cdot |S_1|}{|S_2|})\} \forall i \in S_2. \quad (7.17)$$

### 7.3.3 An affine offset transformation of a user to a tutor

An affine transformation on the edges of each subsequence of a motion synchronisation  $\mathbb{S}$  is applied in order to adjust a tutor's motion data series  $E_T$  to that of  $E_P$ . The proposed Algorithm 13 encompasses assigning the mean pose of the tutor for each landmark type, i.e.  $\bar{P}(E_T, \chi^G_T, \tau)$  (see (5.30) on Section 5.5.2) to each corresponding landmark  $\chi_i \in \chi^G_T$  satisfying  $TYPE(\chi_i) = \tau$  for all  $\tau$ . Then, in order to render the whole motion smooth, the motion data in between landmarks is recalculated applying an affine transformation on the offset of the angular information. The level of adjustment (i.e. whether to make the tutor's level of performance look exactly like the user's) is

**Algorithm 12** Motion data series synchronisation algorithm

---

```

function SynchroniseMotionSeries( $E_T, E_P, \chi^G_T, \bar{\chi}_P$ )    ▷ Synchronise the two
landmark sequences first (i.e., tutor's and user's)
   $\mathbb{S} \leftarrow$  SynchroniseLandmarkSeries( $\chi^G_T, \bar{\chi}_P$ )    ▷ Order the aligned landmarks in
time by the value of the frames of the tutor's landmarks
   $\mathbb{S} \leftarrow$  OrderByFirstElementFrame( $\mathbb{S}$ )    ▷ Empty synchronised motion series to
begin with
   $E_T^{\mathbb{S}(E_P)} \leftarrow \emptyset$ 
  for  $i \leftarrow 1, |\mathbb{S}|$  do
    if  $\mathbb{S}(i, 2) \leq \mathbb{S}(i + 1, 2)$  &  $\mathbb{S}(i, 1) \leq \mathbb{S}(i + 1, 1)$  then    ▷ For each contiguous tuple
( $\chi^G_T(i), \bar{\chi}_P(i)$ ), calculate the user's and tutor's sub-sequences between landmarks ( $\mathbb{SS}_P$  and
 $\mathbb{SS}_T$  respectively)
       $\mathbb{SS}_P \leftarrow \{\mathbb{S}(i, 2) + 1 \dots \mathbb{S}(i + 1, 2)\}$ 
       $\mathbb{SS}_T \leftarrow \{\mathbb{S}(i, 1) + 1 \dots \mathbb{S}(i + 1, 1)\}$ 
      if  $|\mathbb{SS}_P| > 0$  &  $|\mathbb{SS}_T| > 0$  then ▷ Synchronise the two sub-sequences and append
to the motion data series
         $\mathbb{SS} \leftarrow$  SynchroniseSubSequence( $\mathbb{SS}_T, \mathbb{SS}_P$ )
         $E_T^{\mathbb{S}(E_P)} \leftarrow \{E_T^{\mathbb{S}(E_P)}, E_T(\mathbb{SS})\}$ 
      end if
    end if
  end for
  return  $E_T^{\mathbb{S}(E_P)}$ 
end function

```

---

modulated by  $\gamma$ . Therefore, rather than showing the animation of  $M_P$  and  $M_T$  superimposed, the proposed approach uses  $M_P$  and  $\text{AffineOffsetTransformation}(E_P, E_T, \bar{\chi}_P, \bar{\chi}_T, \gamma)$ .

## 7.4 Experimental results

In this section, a series of results drawn from the experimentation explained in Chapter 7 are explored and discussed. First, an evaluation of the trivial algorithm for periodic landmark sequence analysis and its performance is compared against that of the Genetic Algorithm (GA). The results for the best configuration in the landmark refinings stage are also shown, in order to help understand the effect of the methods for periodic landmark sequence analysis. Then, the proficiency achieved on landmark assessment is shown in a similar fashion. Further results show both textual and visual feedback through motion synchronisation between tutor and user and motion adjustment of the performance with the performance level parameter  $\gamma = 0.5$ .

The parameters chosen for each motion class are summarised in Table 7.1. The average overall number of assessments to be inferred ( $G \cdot N_j$ ) expresses the average sample size for evaluation

**Algorithm 13** Motion data series adjustment algorithm

---

```

function AffineOffsetTransformation( $M_1, M_2, \chi_1, \chi_2, \gamma$ )
   $M_A \leftarrow M_1$ 
  for all  $\chi_i \in \chi_1(2 \dots N_T)$  do
     $\tau \leftarrow \text{Type}(\chi_i)$  ▷ For each adjacent landmark...
     $f_i \leftarrow \text{Frame}(\chi_i)$ 
     $f_{i-1} \leftarrow \text{Frame}(\chi_1(i-1))$  ▷ ... and for each involved body joint...
    for all  $j_i \in J$  do ▷ ... calculate the offset between the motion data series...
       $\delta_1 \leftarrow (M_A(f_{i-1}, j) - M_1(f_{i-1}, j)) \cdot \gamma_{j_i}$  ▷ ... in the left edge of the subsequence...
       $\delta_2 \leftarrow (\bar{P}(M_2, \chi_2, \tau, j_i) - M_1(f_i, j_i)) \cdot \gamma_{j_i}$  ▷ ... in the right edge...
       $\Delta \leftarrow (\delta_2 - \delta_1) / (f_i - f_{i-1})$  ▷ ... and per frame
      for all  $s_k \in M_1(f_{i-1} \dots f_i, j)$  do ▷ Apply an affine offset transformation
         $M_A(k, j_i) \leftarrow s_k + (\delta_1 + \Delta \cdot (k - f_{i-1}))$  ▷ Add an increment of  $\Delta$  per frame
      end for
    end for
  end for
  return  $M_A$ 
end function

```

---

TABLE 7.1: Details of parameters of the training samples and testing samples for landmark assessment per exercise, with  $G$ =average number of ground truth landmarks per sample,  $M^A$ =motion data source (where  $E$ =Euler angles and  $Q$ =quaternions),  $N_A$ =number of assessment levels (number of *classes* per body joint) and  $G \cdot N_J$ =average overall number of assessments to be inferred

Exercise	$G$	$M^A$	$N_A$	$G \cdot N_J$
Ankles stretches	13.55	Q	4	54.20
Arms raises	6.65	E	4	26.60
Calves stretches	13.68	Q	4	54.72
Inner thighs stretches	6.84	Q	4	34.20
Shoulders and upper back stretches	9.37	Q	4	37.48
<b>Average</b>	<b>9.78</b>		<b>4</b>	<b>41.44</b>

(i.e., the number of identified landmarks in the landmark identification stage times the number of body joints to be assessed). A number of assessment levels of 4 (i.e., possible values for assessment of 1, 2, 3 or 4) was chosen in order to simplify the training stage. Issues associated with the value of this parameter are discussed in Section 8.3.

### 7.4.1 Landmark assessment

Each exercise sample was manually labelled with an assessment level within the values (1..4) (i.e.,  $N_A = 4$ ). This time, evaluation is based on the ratio of successfully assessed landmarks to

the total number of landmarks. The ratios of false and true negatives and positives for assessed landmarks ( $fn_A$ ,  $fp_A$ ,  $tn_A$  and  $tp_A$ , respectively) are calculated as

$$\begin{aligned}
tp_A(\bar{\chi}, \chi^G) &= \left| \bigcup_{j_i \in J} \{ \chi_i \in \bar{\chi} / \exists \chi_k \in \chi^G / \text{FRAME}(\chi_i) \in [\Lambda_j^1, \Lambda_k^2] \right. \\
&\quad \left. \text{AND AL}(\bar{\chi}_i, j_i) = \text{AL}(\chi_k, j_i) \right\} |, \\
fp_A(\bar{\chi}, \chi^G) &= \left| \bigcup_{j_i \in J} \{ \bar{\chi}_i \in \bar{\chi} / \forall \chi_k \in \chi^G \text{ FRAME}(\bar{\chi}_i) \notin [\Lambda_k^1, \Lambda_k^2] \right. \\
&\quad \left. \text{OR TYPE}(\bar{\chi}_i) \neq \text{TYPE}(\chi_k) \right. \\
&\quad \left. \text{OR AL}(\bar{\chi}_i, j_i) \neq \text{AL}(\chi_k, j_i) \right\} |, \\
fn_A(\bar{\chi}, \chi^G) &= \left| \bigcup_{j_i \in J} \{ \chi_i \in \chi^G / \forall \bar{\chi}_k \in \bar{\chi} \text{ FRAME}(\bar{\chi}_k) \notin [\Lambda_i^1, \Lambda_i^2] \right. \\
&\quad \left. \text{OR TYPE}(\bar{\chi}_k) \neq \text{TYPE}(\chi_i) \right. \\
&\quad \left. \text{OR AL}(\bar{\chi}_k, j_i) \neq \text{AL}(\chi_i, j_i) \right\} |, \\
tn_A(\bar{\chi}, \chi^G) &= N_J \cdot |\bar{\chi}| - tp_A(\bar{\chi}, \chi^G) - fp_A(\bar{\chi}, \chi^G) - fn_A(\bar{\chi}, \chi^G).
\end{aligned} \tag{7.18}$$

Then, given a set of assessed landmarks  $\bar{\chi}$ , precision ( $\pi_A$ ) and accuracy ( $\alpha_A$ ) for assessed landmarks are calculated as

$$\begin{aligned}
\pi_A(\bar{\chi}, \chi^G) &= tp_A(\bar{\chi}, \chi^G) / (tp_A(\bar{\chi}, \chi^G) + fp_A(\bar{\chi}, \chi^G)) \\
\alpha_A(\bar{\chi}, \chi^G) &= (tp_A(\bar{\chi}, \chi^G) + tn_A(\bar{\chi}, \chi^G)) / \\
&\quad (tp_A(\bar{\chi}, \chi^G) + tn_A(\bar{\chi}, \chi^G) + fp_A(\bar{\chi}, \chi^G) + fn_A(\bar{\chi}, \chi^G)).
\end{aligned} \tag{7.19}$$

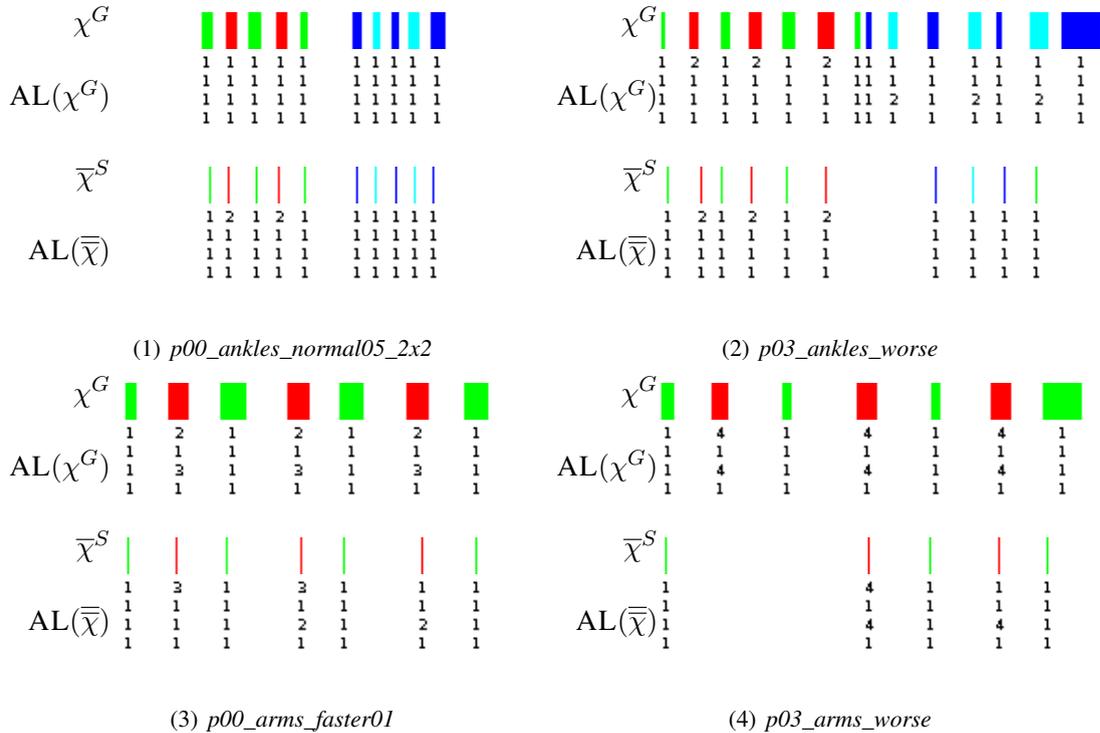
TABLE 7.2: Average precision ( $\pi_A$ ), accuracy ( $\alpha_A$ ) false positives rate ( $fp$ ) and false negatives rate ( $fn$ ) achieved on landmark assessment, with  $TT$  and  $ET$ =Training and execution times, respectively (in s.)

Exercise	$\pi_A$	$\alpha_A$	$fp_A$	$fn_A$	$TT$	$ET$
Ankles	0.90	0.92	0.41	0.41	0.4634	0.0018
Arms	0.89	0.92	0.55	0.55	0.4545	0.0015
Calves	0.88	0.90	0.67	0.67	0.5878	0.0019
Inner thighs	0.94	0.93	0.51	0.51	0.5079	0.0019
Shoulders	0.70	0.89	0.58	0.58	0.3343	0.0011
<b>Average</b>	<b>0.86</b>	<b>0.91</b>	<b>0.54</b>	<b>0.54</b>	<b>0.4658</b>	<b>0.0016</b>

Table 7.2 shows precision, accuracy, false positive and false negative rates achieved with the proposed approach for landmark assessment, including the compromise interval criteria for precision calculation with  $\Delta_\Lambda = 0.10$  (i.e. maximum 10% motion drift between landmarks). Also, 10-fold cross-validation with 5 repetitions were carried out for each motion class, in multi-combination with 10-fold cross validation for landmark identification and landmark refining.

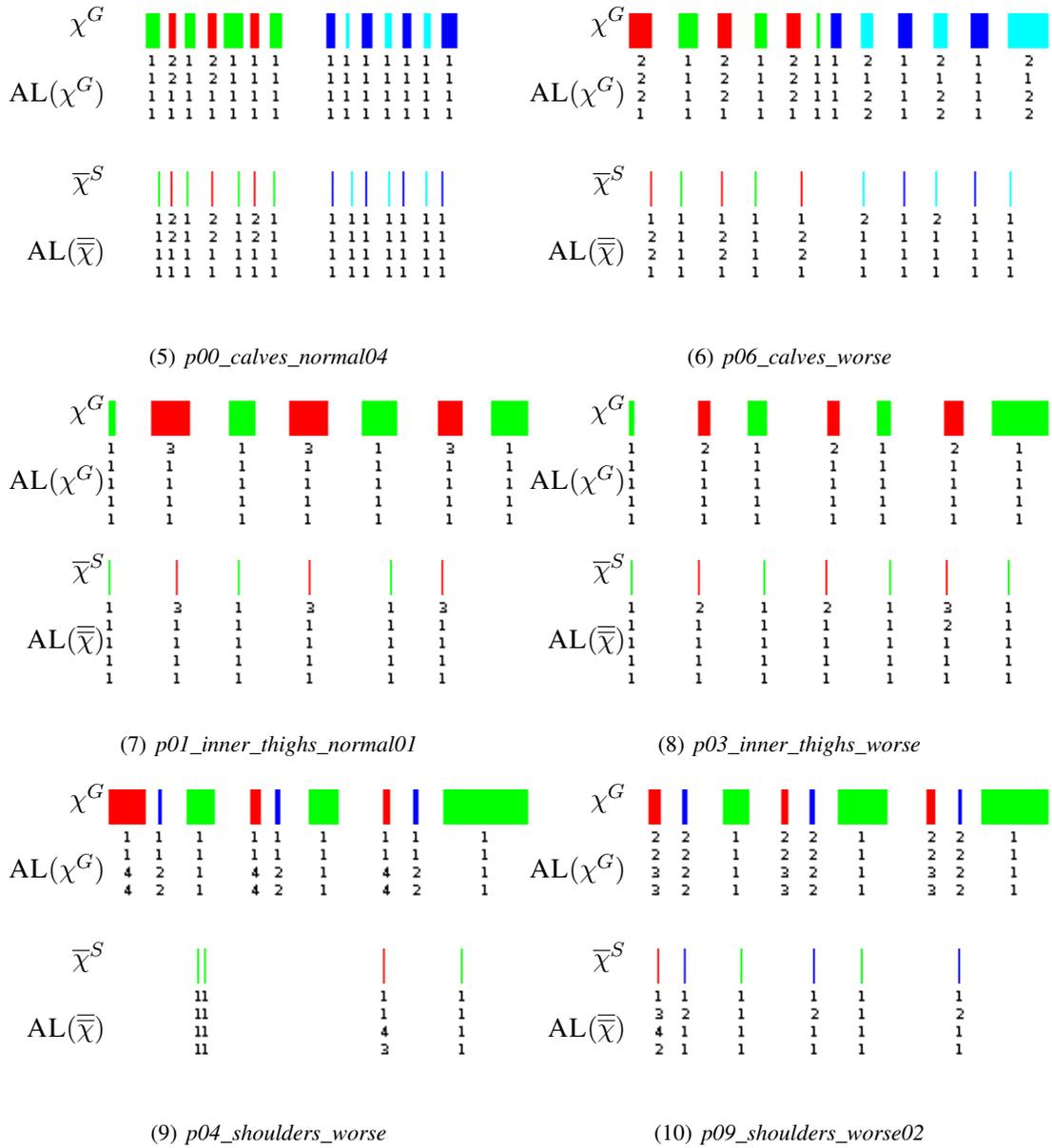
Figure 7.7 shows a graphical representation of a number of different results for landmark identification and landmark assessment. The ground truth landmarks ( $\chi^G$ ) refer to the compromise intervals of the ground truth landmarks of the performance being assessed. ground truth assessment ( $AL(\chi^G)$ ) refers to the assessment levels of the ground truth landmarks as shown in the figure. Likewise, the results of periodic landmark sequence analysis ( $\bar{\chi}^S$ ) show the improved landmark sequences after landmark period analysis.

FIGURE 7.7: Representation of results for periodic landmark sequence analysis and landmark assessment. Ground truth landmarks ( $\chi^G$ ) and periodic landmark sequence analysis ( $\bar{\chi}^S$ ) represented with vertical markers of different colours, each of which represents a ground-truth, identified or assessed landmark type, respectively, and their assessment levels ( $AL(\chi^G)$  and  $AL(\bar{\chi})$  respectively), each number standing for an assessment level inferred on an involved body joint, in vertical order (from top to bottom) as per Table 5.2



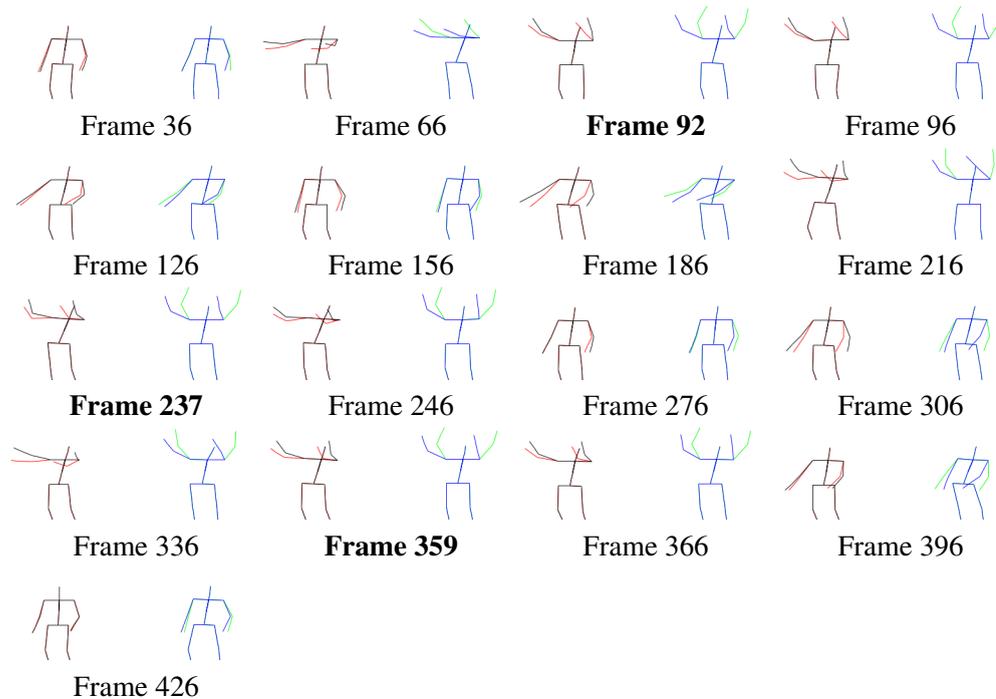
Continued on next page

FIGURE 7.7: Representation of results for periodic landmark sequence analysis and landmark assessment (continued from previous page)



The assessment levels of the assessed landmarks ( $AL(\bar{\chi})$ ) show the identified landmarks and the inferred assessment levels. Colours represent landmark TYPES, i.e., RoM extrema. Each assessment level corresponds, in respective order, to an involved body joint. In this case, the smaller the value of the assessment level, the better the performance. Further examples of landmark assessment and periodic landmark sequence analysis can be found in Appendix B.

FIGURE 7.8: Generated assessment and adjustment for *p00\_arms\_worse03*. Sequence of the reconstructed skeletons of the user's performance (in red) and its adjustment to the tutor's (in black) with  $\gamma = 0.5$ . Additionally, the synchronised animation of the tutor's demonstration (in green) and its adjustment to the user (in blue) with  $\gamma = 0.5$ . The animation is to be followed from left to right and top-bottom



Generated feedback:

- Frame 92: “*On this occasion, you are not rising your left arm. Furthermore, you are not rising your right arm. Look at the instructor and try to reach up with your arms on top of your head.*”
- Frame 237: “*Now, once more, you are not rising your left arm. Once more, you are not rising your right arm.*”
- Frame 359: “*This time, again, you are not rising your left arm. Again, you are not rising your right arm.*”

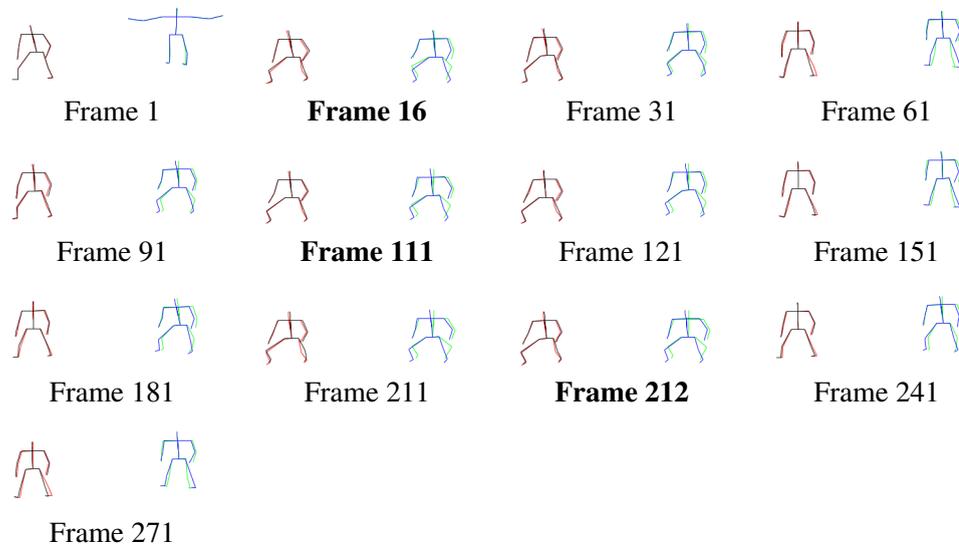
#### 7.4.2 Generated assessment feedback

Figure 7.8 shows a graphical representation of the visual feedback delivered to the user after performance of the *Arm raises* exercise. It is an animation of the user's performance, along with its motion adjustment to that of the tutor, with a performance level parameter value of  $\gamma = 0.5$ . In this figure, synchronisation of the tutor's performance, by virtue of the synchronised landmarks, can be observed. Also, an adjustment of the tutor's performance to that of the user is shown along.

Together with the animation, the generated textual feedback is shown, linked to the relevant time frames.

Notice the difference between the appearance of both user and tutor in a number of poses (nominally, those matching the identified and ground truth landmarks). Motion adjustment allows to clearly transmit the idea that the performance could be much better, but (at least, with a value for  $\gamma$  close to 0.5) it will never reflect a big difference with respect to the reference. This way, behaviour of the user, in terms of encouragement, can be explored in a natural way.

FIGURE 7.9: Generated assessment and adjustment for *p04\_inner\_thighs\_worse*. Sequence of the reconstructed skeletons of the user's performance (in red) and its adjustment to the tutor's (in black) with  $\gamma = 0.5$ . Additionally, the synchronised animation of the tutor's demonstration (in green) and its adjustment to the user (in blue) with  $\gamma = 0.5$ . The animation is to be followed from left to right and top-bottom



Generated feedback:

- Frame 16: “*On this occasion, your left knee needs to be bent a little bit more. And your right knee needs to be bent a little bit more. Try ducking lower.*”
- Frame 111: “*This time, once more, your left knee needs to be bent a little bit more. Again, your right knee needs to be bent a little bit more.*”
- Frame 212: “*Now, again, your left knee needs to be bent a little bit more. Once more, your right knee needs to be bent a little bit more. In addition, your left hip needs to be stretched a little bit more. In addition, your right hip needs to be stretched a little bit more. Try moving your left foot further apart from your right foot. In addition, try moving your right foot further apart from your left foot.*”

Furthermore, the textual feedback helps understand the situation, even in cases when visual interpretation is not straight forward. This is the case shown in Figure 7.9. A combination of feedback and tips (the later only been shown once) is uttered to the user and the animation stops, giving the user a chance to study the information. Further examples of motion adjustment and motion synchronisation animations can be found in Appendix D.

## 7.5 Conclusions

In this chapter, a novel method for automatic, human-like landmark assessment production of an exercise's performance upon previously identified landmarks has been presented. The approach takes into account restrictions on landmark sequence repetitiveness, landmark sequence symmetry and *inaccuracy* of the identified landmarks.

A method for semi-supervised, machine learning-based matched landmark feature modelling and inferring of numerical assessment has been discussed. Furthermore, a method based on a context-free meta-grammar to produce human-understandable feedback has also been presented.

All the above is displayed to the user as a graphical animation of their skeletal data series combined with an motion adjustment calculated as an affine transformation of the original. An algorithm for pair-wise matching of landmarks (motion synchronisation) given two sequences belonging to two different individual performances is introduced. Finally, the inclusion of a performance level parameter ( $\gamma$ ) for level of performance tracking through the entire life cycle of the system is introduced. The usage of  $\gamma$  for motion adjustment has been suggested. Results on the evaluation of the methods introduced in this chapter are shown and discussed in Section 7.4.

Results show that periodic landmark sequence analysis successfully improves the average structural arrangement of a sequence of identified landmarks, in terms of a lower rate of false positives, which in turn improves motion synchronisation and feedback production.

Furthermore, landmark assessment proved to be accurate when compared with the ground truth. Low false positives and low negative rates show the precision and accuracy of both the key body pose identification (encompassing landmark identification, landmark refining and periodic landmark sequence analysis) and the landmark assessment stages. Worded feedback is produced accordingly and a series of animations are created in order to show the validity of the landmark assessment, motion synchronisation and motion adjustment algorithms.

## Chapter 8

# Limitations of the approach and technical issues

In this chapter, a number of issues of the approach are discussed. Limitations of use of the Kinect camera are reviewed. Issues present in the choice of the training sets depending on various system parameters are discussed.

### 8.1 Accuracy of the Kinect camera: self-occlusion and orientation of the individual

A number of reports [57, 84, 106] point out the limitations of the Kinect camera in terms of accuracy of the reconstructed joint coordinates and angles. Orientation of the user with respect to the location of the camera is an important factor, as self-occlusion of body limbs results in incorrect reconstruction of the body joints.

Occlusion will typically occur when an object, another individual, or even part of the individual's body gets in the way between a specific part of the body and the Kinect camera. When another part of the body acts as this extraneous agent, self-occlusion occurs. This behaviour is analysed in [84] for rehabilitation exercises mainly directed to elderly people. Often, a drift of a few angles away from the frontal orientation to the camera result in self-occlusion. Examples include a knee or a hip occluded by the opposite leg or abdomen respectively; shoulder occluded by the head; parts of the body hidden behind objects like chairs, tables or walls.

Take as an example the animation of the subject *p04\_arms\_normal02* shown in Figure 8.1. In frames #11 and #12 the right arm appears in a horizontal position. In reality the individual had moved it back to its relaxed, vertical position (as shown by the black coloured superimposition). This is due to a failed motion capture reconstruction of the joints.

FIGURE 8.1: Illustration of a motion capture error. Sequence of the reconstructed skeletons (every 10 frames) of *p04\_arms\_normal02* performance (in red) and its motion adjustment to the tutor's (in black). The animation is to be followed from left to right and top-bottom

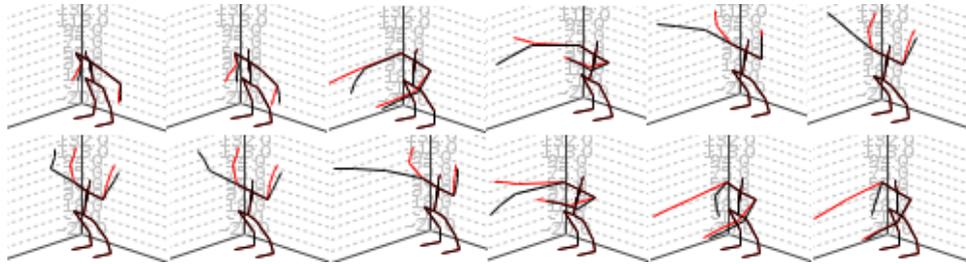


Figure 8.2 depicts the origin of this failed reconstruction. The patch in light blue surround the values of the Euler angles near the moment when the individual recovers to the relaxed pose. If compared with other instances of the same pose (in light blue marker), one can notice that the value of an Euler angle (in this case, RIGHT\_SHOULDER's *bank* and *heading*) is not recovered to the original value.

FIGURE 8.2: Euler angles of a failed recovered sample (*p04\_arms\_normal02*). The data represent only the RIGHT\_SHOULDER and LEFT\_SHOULDER's motion. The animation depicted in Figure 8.1 corresponds with the interval highlighted with a grey patch. The problematic Euler data is highlighted by a dashed red marker and similar poses are surrounded by dashed light blue markers



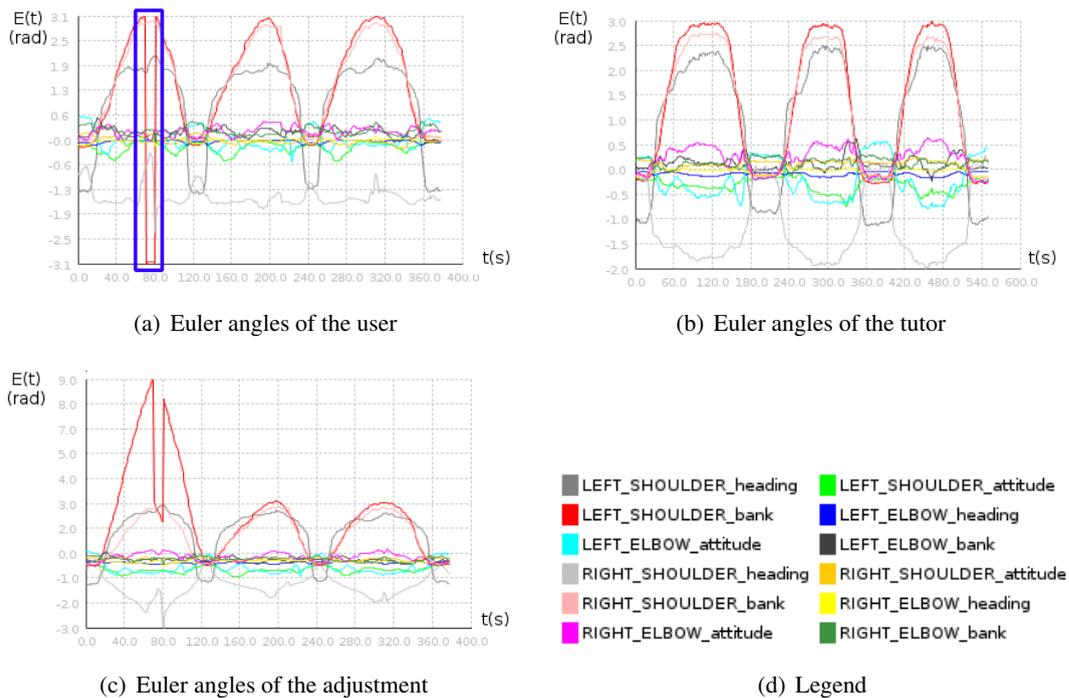
## 8.2 Discontinuities in the Euler angles due to gimbal lock

One of the issues of using Euler angles is that, in many cases, more than one unique succession of angles *heading*, *attitude* and *bank*, (or  $(\Theta, \Phi, \Psi)$ ) –similar, conceptually, to the *yaw*, *pitch* and *roll* angles used in robotics applications– may define the same pose. This is a common phenomenon when using Euler angles which is known as the *gimbal lock*, that appears when

the axes of two of the angles get aligned in space, causing the system to “lose” one degree of freedom. This causes discontinuities in the signal, as, in near-gimbal-lock areas, the values of one of the angles will switch from the positive hemisphere to the negative and vice-versa. It is a very common issue present on 3D human motion recovery [56, 69, 90, 94].

When gimbal lock occurs in the vicinity of a landmark, the method developed in Section 7.3 for adjustment of the user to the tutor fails to reconstruct realistic data. This is originated in Algorithm 13, when, in lines 8 and 9, the values of  $\delta_1$  and  $\delta_2$ , respectively, are calculated as the difference between the Euler angles value of the tutor and the user. When either  $E_T(t, j)$  or  $E_P(t, j)$  fall within a gimbal lock area, the values of either  $\delta_1$  or  $\delta_2$ , or both, will drift away.

FIGURE 8.3: Illustration of gimbal lock in motion adjustment for the *p01\_arms\_normal01* sample



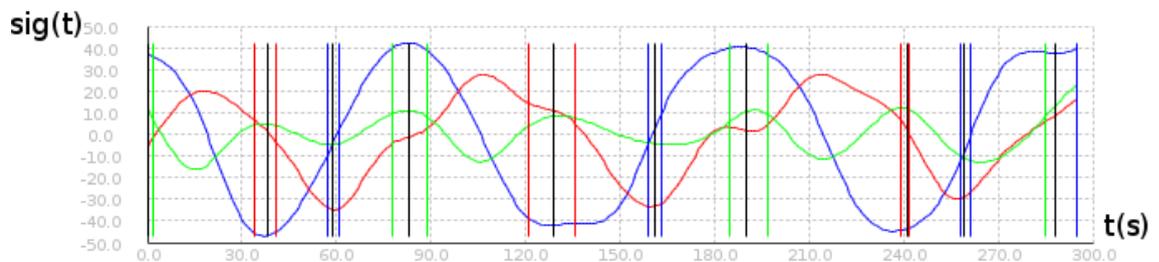
Take as an example the adjustment of the user’s sample labelled as *p01\_arms\_normal01* in the *Arms* exercise. The values of the Euler angles of the involved body joints (LEFT\_ELBOU, RIGHT\_ELBOU, LEFT\_SHOULDER and RIGHT\_SHOULDER) are shown in Figure 8.3 (a). The interval affected by the gimbal lock is highlighted in the blue box. Although the abnormal pattern seemed to be caused by noise, the motion capture framework actually reconstructed the values of the LEFT\_SHOULDER on a different reference frame, so that the skeletal information can be recovered smoothly.

On the other hand, the Euler angles retrieved by the motion capture framework for the tutor of the same exercise (*p00\_arms\_normal01*) are shown in Figure 8.3 (b). Note how the interval corresponding to the same interval in Figure 8.3 (a) (as per synchronisation of both user's and tutor's landmarks using Algorithm 11 shown in Section 7.3.1) does not present this issue. The resulting Euler angles after adjusting *p01\_arms\_normal01* to *p00\_arms\_normal01* by Algorithm 13 are shown in Figure 8.3 (c). Note the big drift in the gimbal lock area.

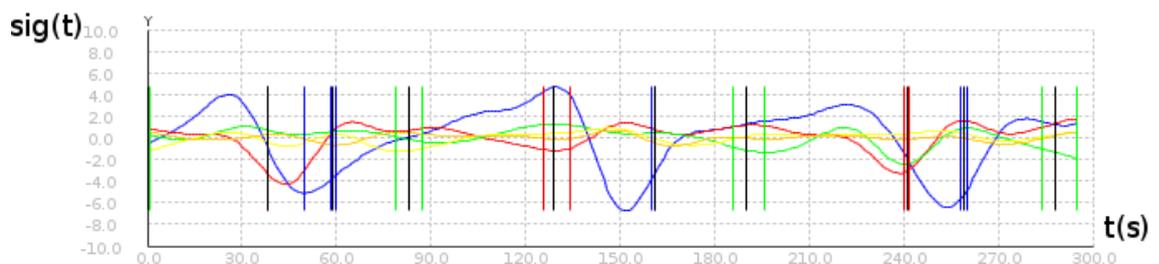
Several approaches exist to tackle the gimbal lock effect on posture recovery, like using a different reference for rotation [56] or using quaternions [69] rather than Euler angles. The latter introduces representation of angles in a different format, but it seems like the most feasible solution, since quaternions are the only format provided by the (discontinued) OpenNI 2.0 platform (whereas OpenNI 1.0, providing Euler angles, was used in this research).

### 8.2.1 Unsuitable motion signature

FIGURE 8.4: Comparison of the calculated motion signature between absolute body joint positions and Euler angles of the *p01\_shoulders\_normal01* sample. Each wave represents a different *eigenvalue*. Landmarks are depicted as vertical markers and the stroke colour indicates their landmark type



(a) Motion signature calculated from the absolute joint positions

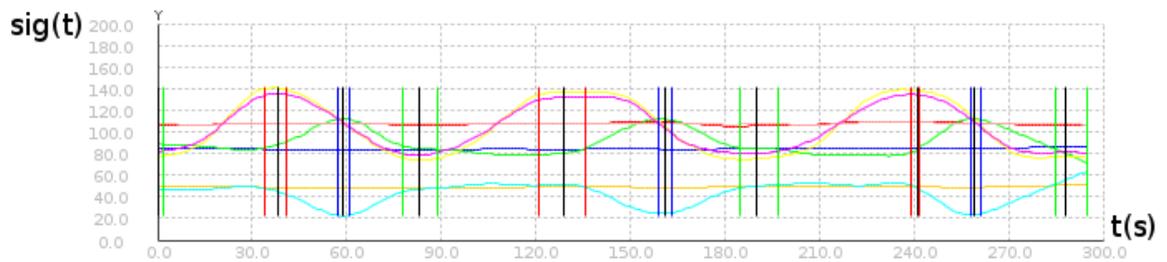


(b) Motion signature calculated from the Euler angles

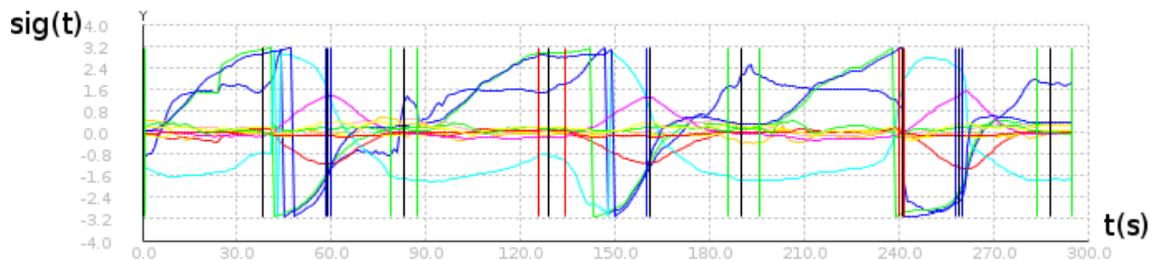
Discontinuities on the Euler angles extracted with the motion capture framework present significant issues for the calculation of landmark features for landmark identification. As shown in Section 5.2, these are extracted from the motion signature *sig*, which in turn are calculated as a

subset of the *eigenvalues* of the Euler angles. In order to calculate the motion signature for these exercises, the absolute positions  $P$  of the involved joints are used as input data, rather than the Euler angles, for almost all motion classes (see Table 5.1 in Section 5.6). A representation of the different motion signatures calculated from both Euler angles ((b)) and absolute positions ((a)) is shown in Figure 8.4.

FIGURE 8.5: Absolute positions and Euler angles of the *p01\_shoulders\_normal01* sample. Each wave represents a different body joint and Degree of Freedom (DOF). Landmarks are depicted as vertical markers and the stroke colour indicates their landmark type



(a) Absolute joint positions



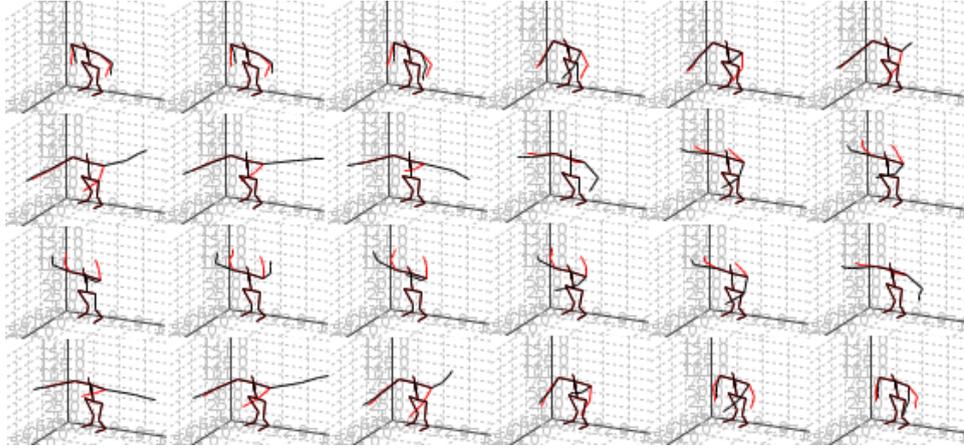
(b) Euler angles

Observe how the values of the *eigenvalues* differ greatly for landmarks of the same type for the motion signature calculated from the Euler angles (Figure 8.5 (b)), e.g. red stroke colour markers for the first Principal Component (PC), blue for the second PC, etc. This introduces a high degree of variance and may cause overfitting of the system. Absolute positions (Figure 8.5 (a)) show a higher consistency in the data features.

## 8.2.2 Unnatural adjustment reconstruction

An illustration of the effect of gimbal lock after skeletal reconstruction for adjustment is shown in Figure 8.6. The result is a misleading representation of the motion of the left arm, showing impossible swinging movement of the shoulder in the transition between landmarks.

FIGURE 8.6: Illustration of the Gimbal Lock issue in motion adjustment. Sequence of the reconstructed skeletons (every 5 frames) of user's performance (in red) and its adjustment to the tutor's (in black). The data is the same used in Figure 8.3. The animation is to be followed from left to right and top-bottom



### 8.3 Escalation of training samples

Another issue in the field of machine learning is estimating the optimal size of the training set. If the system is trained with few samples some classes may be unequally trained, if trained at all. On the other hand, too many training samples may result in overfitting.

This issue affects the performance of both the landmark identification and landmark assessment learning stages, but it is perhaps more obvious in the latter, as the number of classes (i.e., assessment levels) may vary greatly. To evaluate the system, a value of  $N_A = 4$  (i.e., number of assessment levels) was chosen for simplicity purposes. However, in a realistic scenario, this value may need to be greater, in order to provide a greater range of assessment expressiveness.

Let  $K$  be an empirically estimated constant value standing for the number of observations (both positive and negative) needed to train the system optimally for a single body joint, landmark type and assessment level. Then, the overall estimated number of training observations  $\mathbb{T}r$  needed for an exercise to be optimally trained by the system will need to be

$$\mathbb{T}r \approx N_J \cdot T \cdot N_A \cdot K, \quad (8.1)$$

where  $N_J$  is the number of involved body joints and  $T = |\tau \in \text{TYPE}(\chi^G)|$ . With the average values of  $N_J = 4$ ,  $T = 4$  and  $N_A = 4$  in the dataset, the approximate value of  $\mathbb{T}r = 64 \cdot K$ . (8.1) is derived from the fact that each modelled class stands for an individual landmark type, but is

calculated upon the physical appearance of each involved body joint, i.e., the values of each body joint's data series. Furthermore, there are  $N_A$  class values.

A single sample may contain several observations. Each sample will contribute with a 50% of negative observations, i.e.,  $\mathcal{O}/\text{TYPE}(\mathcal{O}) = \text{none}$  and 50% of positive observations or ground truth landmarks, i.e.,  $\chi^G$ .  $\mathbb{T}r$  can then be low-bounded as

$$\mathbb{T}r \geq \frac{N_J \cdot T \cdot N_A \cdot K}{2 \cdot |\chi^G|}, \quad (8.2)$$

With an average value of  $|\chi^G| = 10$ ,  $\mathbb{T}r \geq 3.2 \cdot K$ . This is the best case since a random sample captured from an individual will normally consist of a succession of identical poses per landmark type, i.e.

$$\mathbb{T}r \approx \frac{N_J \cdot T \cdot N_A \cdot K}{T} = N_J \cdot N_A \cdot K, \quad (8.3)$$

being  $\mathbb{T}r = 16 \cdot K$  with the average values.

If, as previously pointed out, one wants to increase the number of assessment levels,  $N_A$  will become a parameter, so that  $\mathbb{T}r = 4 \cdot N_A \cdot K$ . This may introduce an escalation factor for the training set, particularly for high values of either  $K$  or  $N_A$ , or both.

## 8.4 Conclusions

This chapter discussed a series of limitations of the approach, mainly related to issues present in the Motion Capture (MoCap) technology used for this research. Lack of accuracy in the Kinect camera is a well known issue that is repeatedly referred to in the literature. This can have negative effects in evaluation scenarios, especially when present in training samples.

Furthermore, gimbal lock is always present in scenarios using the Euler angles reference system. Its effect is often an unnatural adjustment reconstruction and an unsuitable motion signature. It is suggested the use of quaternions in order to suppress this phenomenon, sacrificing the readability of the data.

Finally, the issue of training sets size escalation as an effect of a highly versatile choice of parameters is presented and discussed. In short, the higher the number of landmark types and assessment levels, the bigger the training set will be.

## Chapter 9

# Conclusions and future work

A novel human motion alignment methodology for stretching exercises aimed at elderly people was presented. This consists of extracting a set of features from key frames of a dimensionality-reduced data series of the smoothed input from a Kinect device. Although the motion data from the device was rather unreliable, re-sampling and Principal Component Analysis (PCA) have produced usable data. The proficiency achieved using different configurations was tested and compared. The configurations were parametrised in virtue of: which learning algorithm was used for landmark identification; whether the dimensionality of the multivariate input data was reduced; whether frame-wise features were extracted; whether landmark refining (by clustering) was performed and, if so, which cluster selection algorithm was used.

The best configuration was compared against the Dynamic Time Warping (DTW) and Hierarchical Aligned Cluster Analysis (HACA) algorithms, showing significantly better performance in the scenarios. Specifically, approaches that consider the whole data spectrum, like HACA, are shown not to be suitable for exact segmentation by key body poses. The identification of key body poses allow matching a sample motion with a reference, ground truth sample (the tutor) for performance assessment. Furthermore, unlike approaches that follow a similar method to DTW, the learning and clustering methods presented in this paper achieve consistent performance in certain motion classes where the key body poses in the motion object of analysis do not exactly match the expected pattern. Specifically, the approach does not limit individuals to performing a fixed number of repetitions per stretch/reach and exercising a particular side of the body first (left/right). This a novel feature of the proposed approach on motion analysis.

## 9.1 Contributions

In this thesis, a number of original contributions towards semi-supervised key body poses identification in and assessment of motion capture samples have been presented.

### **A method for semi-supervised key body pose identification based on frame-wise learning**

Motion alignment for automatic assessment production of fitness performances has been addressed on the basis that the performance itself may not correspond exactly with a predefined reference. As a foundation step towards motion alignment, common key body pose identification is solved first. It is assumed that the common format of these exercises include a number of repetitions of the same body limb stretch and, in some cases, further repetition with the limbs of the opposite side of the body. Specifically, discrepancies in the *repetitiveness* and *symmetry* are considered. In order to tackle this, a flexible, frame-wise key body pose (or *landmark*) learning method, based on machine learning and clustering algorithms, has been developed and evaluated. Furthermore, experimentation with originally conceived *frame-wise features* showed significant improvement in frame classification. These features, calculated from the original, dimensionality-reduced motion data, model the relation between noise and signal.

### **A genetic algorithm for extraneous member removal in periodic sequences**

periodic landmark sequence analysis is performed in order to remove false positives (i.e., out-of-sequence identified landmarks). Two methods have been tested: a trivial algorithm and a Genetic Algorithm (GA). The GA applies prior knowledge on periodic patterns in order to produce hypotheses of extraneous members. The approach is evaluated using two different domains, one of them being the refinement of periodic sequences of identified key body poses with the presence of false positives. Furthermore, synthetic data of long periodic strings are also evaluated. The GA compares favourably against classical string approximate matching algorithms like Needleman-Wunsch and Smith-Watson on the targeted domains. Results show high versatility and potential use on applications requiring approximate periodic pattern matching like intrusion detection and unusual behaviour analysis. Optimal parameters of the GA and some future research guidelines are commented, including the application on open challenges in abnormal pattern recognition

**A method for motion alignment based on key body poses**

Motion alignment between a previously unseen motion data sample (that of the user) and a reference or exemplary performance of the exercise (performed by the tutor) has been attempted by matching of common key body poses (i.e., landmarks and their landmark types). It is assumed that the two sequences of landmarks present dissimilarities for the reasons given in the previous point. It is not assumed that the key body poses are fully recovered, as the landmark identification method proved not to be perfect. An algorithm to synchronise two motion data samples upon landmark matching has also been developed and tested. This can be applied to motivational research of the users in real world scenarios, conveying visual feedback with superimposed and synchronised animations of both user's performance and that of the tutor.

**An algorithm for semi-supervised key body pose assessment based on matched features**

A method based on machine learning and feature parsing for automatic assessment production of key body poses has been developed and evaluated in this thesis. It describes how the learning features are calculated upon synchronised series of landmarks. A multi-label, multi-class learning algorithm is trained with these features, along with the *type of body pose*, to infer a numerical value for each key body pose and body joint. A method to translate these values into human-understandable messages based on a grammar that parses these features has also been described and tested.

**An algorithm for adjustment of two aligned motion sequences**

The final contribution of this thesis is a method for adjusting the skeletal animation of user's performance to that of the tutor, based on the Euler angles information of the synchronised poses. An affine transformation of the angular data series is performed over the involved joints, leaving the rest untouched. The purpose is to contribute towards the visual feedback by showing a natural representation of the user that is closer to the tutor, by maintaining the original appearance. This addresses the problem of having two individuals executing an exercise on a similar –but different– orientation towards the motion capture device and with different body scales. The motion adjustment is performed as a function of the performance level parameter that represents the average level of performance of the user through time.

## 9.2 Analysis of issues and strengths

The approach has some drawbacks. Firstly, it relies on stable input data with a low noise component and good accuracy. Empirical work showed that the data extracted with Kinect is not accurate. Therefore, more work has to be done to improve motion capture. Moreover, self-occlusion of body limbs produces only moderately satisfactory outcome on landmark identification, producing both false positives and negatives (i.e., non-existent and missed landmarks, respectively). This becomes even more problematic when performance measurements, like speed, are to be calculated based on *landmark* sequencing information. However, the studied exercises involve repetitions so it is very likely that at least one landmark will be identified.

The true potential of using frame-wise information, rather than the complete motion information, lies in the possibility of discovering learnt body poses regardless of the repetitiveness and symmetry of the performance. State-of-the-art temporal segmentation and alignment methods generally expect that an unseen observation shares, to a certain degree, a series of features with the known truth (nominally, the number of primitives or repetitions of each kind of stretch). The proposed approach is not affected by this constraint.

Assessment of a body pose in objective terms is a complex task and personal bias may be introduced. For example, if the judgement is done through visual comparison of a representation of body joints then not every angle of perspective may be covered. Also, one may give more importance to factors like balance or posture than to the range of stretch itself. A further source of complexity is when all three degrees of freedom of each joint are taken into account. Likewise, the chosen motion sensing framework does not keep a reliable control over impossible poses and tight joint movement restrictions.

For this project, the motion capture resources were chosen based on affordability and ease-of-use criteria. It is the belief of the author that using a more advanced motion capture dataset (e.g. Carnegie Mellon University Motion Capture Database [21]) should give better results.

## 9.3 Future work

An area of future work is to improve the success rate of the method for landmark identification. A possible research line would be to perform sequence analysis in order to remove incompatible identified landmarks based on the cyclic nature of the sequences (i.e., repetitiveness and

symmetry).

Secondly, issues regarding accuracy of the Motion Capture (MoCap) system need to be tackled. The new Kinect v2 camera has not been tested in this thesis and may give better results. Also, using quaternions rather than Euler angles may improve the quality of the key body pose identification and assessment features.

Case studies of real-world applications are of enormous importance, as they provide substantial feedback on the proficiency of the system. Developing an application for real-time motion capture, analysis and assessment that is capable of functioning in places like nursing homes or physical rehabilitation centres should also be part of future research.

One of the future key research focuses will be the detection of unusual activities in an event sequence, given a pattern. Activities of Daily Living (ADLs [87]) are a good example of available data to be tested. Changes in behavioural patterns of individuals may give important information that can be used e.g. for terrorism prevention or intrusion detection. However, because every daily subsequence is different, the approach cannot be evaluated against a ground-truth. The lack of a baseline to evaluate the performance of the proposed approach means that further research and development should be carried out in order to use  $GA^K$  in completely unsupervised contexts. This may involve conceiving a heuristic for fitness calculation that does not depend on any *a-priori* assumption. The rate of false positives (i.e., missed extraneous members) may be further lowered by using a better fitness function or an improved mutation operator. Future work will also focus on enhancing the way assessment is produced, like translating every tuple  $A(\text{FRAME}(i), j)$  into human-understandable language.

Finally, the assessment framework needs to be expanded in order to become more versatile. For instance, more assessment levels should be supported in an effective way and a more advanced expert system for message composition upon these values needs to be developed. Work on ontologies for meta-data definition and a study on real-world assessment practices in rehabilitation should be carried out.

# List of Publications

## Conference papers

P. Fernández de Dios, P. W. H. Chung, and Q. Meng, “Semi-automatic, landmark-based feedback generation for stand-up exercises”, in *Computational intelligence in information systems* (Springer, 2015), pp. 285–294.

P. Fernández de Dios, Q. Meng, and P. W. H. Chung, “A machine learning method for identification of key body poses in cyclic physical exercises”, in IEEE International Conference on Systems, Man and Cybernetics (SMC 2013) (Oct. 2013), pp. 1605–1610.

## Journal papers

P. Fernández de Dios, P. W. H. Chung, and Q. Meng, “Landmark-based methods for temporal alignment of human motions”, *IEEE Computational Intelligence Magazine* **9**, 29–37 (May 2014).

## To be submitted

P. Fernández de Dios, Q. Meng, and P. W. H. Chung, “An unsupervised GA method for extraneous member removal in periodic sequences”, *IEEE Transactions on Cybernetics*, to be submitted.

# List of References

- 1 G. L. Alexander, T. C. Havens, M. Skubic, M. Rantz, J. M. Keller, and C. Abbott, “Markerless human motion capture-based exercise feedback system to increase efficacy and safety of elder exercise routines”, *Gerontechnology* **7**, 67 (2008).
- 2 S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs”, *Nucleic acids research* **25**, 3389–3402 (1997).
- 3 H. Bao and Z. M. Liu, “Video-based human motion analysis”, *Advanced Materials Research* **403**, 2593–2597 (2012).
- 4 L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains”, *The annals of mathematical statistics* **37**, 1554–1563 (1966).
- 5 B. Ben-Horin, *Introducing openNI*, <http://www.openni.org/>, Accessed: 2012, 2011.
- 6 G. Benson, “Tandem repeats finder: a program to analyze dna sequences.”, *Nucleic acids research* **27**, 573 (1999).
- 7 L. Bergroth, H. Hakonen, and T. Raita, “A survey of longest common subsequence algorithms”, in *String processing and information retrieval, 2000. spire 2000. proceedings. seventh international symposium on (IEEE, 2000)*, pp. 39–48.
- 8 D. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series”, in *KDD workshop, Vol. 10, 16 (1994)*, pp. 359–370.
- 9 T. W. Bickmore, D. Schulman, and C. L. Sidner, “A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology”, *Journal of biomedical informatics* **44**, 183–197 (2011).

- 10 E. Blanchard, P. Chalfoun, and C. Frasson, “Towards advanced learner modeling: discussions on quasi real-time adaptation with physiological data”, in Seventh IEEE International Conference on Advanced Learning Technologies (ICALT) (2007), pp. 809–813.
- 11 A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates”, Transactions on Pattern Analysis and Machine Intelligence **23**, edited by IEEE, 257–267 (2001).
- 12 L. W. Boyette, A. Lloyd, S. Manuel, J. E. Boyette, and K. V. Echt, “Development of an exercise expert system for older adults”, Journal of rehabilitation research and development **38**, 79–92 (2001).
- 13 J. Brekelmans, *Brekel kinect*, [http://www.brekel.com/?page\\_id=155](http://www.brekel.com/?page_id=155), Accessed: 2012, 2011.
- 14 M. B. Brown and A. B. Forsythe, “Robust tests for the equality of variances”, Journal of the American Statistical Association **69**, 364–367 (1974).
- 15 E. Brox, L. F. Luque, G. J. Evertsen, and J. E. G. Hernández, “Exergames for elderly: social exergames to persuade seniors to increase physical activity”, in 5th international conference on pervasive computing technologies for healthcare (pervasivehealth) (2011), pp. 546–549.
- 16 A. Bruderlin and L. Williams, “Motion signal processing”, in Proceedings of the 22nd annual conference on computer graphics and interactive techniques (ACM, 1995), pp. 97–104.
- 17 H. Bunke and U. Bühler, “Applications of approximate string matching to 2d shape recognition”, Pattern recognition **26**, 1797–1812 (1993).
- 18 F. Buttussi and L. Chittaro, “MOPET: A context-aware and user-adaptive wearable system for fitness training”, Artificial Intelligence in Medicine **42**, 153–163 (2008).
- 19 F. Buttussi, L. Chittaro, R. Ranon, and A. Verona, “Adaptation of graphics and gameplay in fitness games by exploiting motion and physiological sensors”, in Smart graphics (Springer, 2007), pp. 85–96.
- 20 T. Bylander and L. Tate, “Using validation sets to avoid overfitting in AdaBoost”, in Flairs conference (2006), pp. 544–549.
- 21 *Carnegie mellon university motion capture database*, <http://mocap.cs.cmu.edu/>, Accessed: 2015, 2015.

- 22 Y.-J. Chang, S.-F. Chen, and J.-D. Huang, “A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities”, *Research in developmental disabilities* **32**, 2566–2570 (2011).
- 23 S.-P. Chao, C.-Y. Chiu, S.-N. Yang, and T.-G. Lin, “Tai chi synthesizer: a motion synthesis framework based on postures and motion instructions”, *Computer Animation and Virtual Worlds* **15**, 259–268 (2004).
- 24 C. Cortes and V. Vapnik, “Support-vector networks”, *Machine Learning* **20**, 273–297 (1995).
- 25 T. F. Cox and M. A. Cox, *Multidimensional scaling* (CRC Press, 2010).
- 26 J. T. Coyle et al., “Use it or lose it-Do effortful mental activities protect against dementia?”, *New England Journal of Medicine* **348**, 2489–2490 (2003).
- 27 M. Crochemore, L. Ilie, and W. Rytter, “Repetitions in strings: algorithms and combinatorics”, *Theoretical Computer Science* **410**, 5227–5235 (2009).
- 28 E. E. Cureton and R. B. D’Agostino, *Factor analysis: an applied approach* (Psychology Press, 2013).
- 29 K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering”, *Image Processing, IEEE Transactions on* **16**, 2080–2095 (2007).
- 30 A. Dasgupta and Y. Nakamura, “Making feasible walking motion of humanoid robots from human motion capture data”, in *IEEE international conference on robotics and automation*, Vol. 2 (1999), pp. 1044–1049.
- 31 L. Davis et al., *Handbook of genetic algorithms*, Vol. 115 (Van Nostrand Reinhold New York, 1991).
- 32 H. Debar, M. Becker, and D. Siboni, “A neural network component for an intrusion detection system”, in *Research in security and privacy, 1992. proceedings., 1992 IEEE computer society symposium on* (IEEE, 1992), pp. 240–250.
- 33 N. Delson and H. West, “Robot programming by human demonstration: adaptation and inconsistency in constrained motion”, in *IEEE international conference on robotics and automation*, Vol. 1 (1996), pp. 30–36.

- 34 A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38 (1977).
- 35 M. Denkinge, T. Nikolaus, C. Denkinge, and A. Lukas, “Physical activity for the prevention of cognitive decline”, *Zeitschrift für Gerontologie und Geriatrie* **45**, 11–16 (2012).
- 36 Feng Zhou, *Software for aligned cluster analysis*, [http://www.f-zhou.com/tc\\_code.html](http://www.f-zhou.com/tc_code.html), Accessed: 2015, 2015.  
  
P. Fernández de Dios, Q. Meng, and P. W. H. Chung, “A genetic algorithm for extraneous member removal in periodic sequences”, in Genetic and evolutionary computation conference (GECCO-2015) (submitted on Jan 2015).
- 37 E. Flores, G. Tobon, E. Cavallaro, F. I. Cavallaro, J. C. Perry, and T. Keller, “Improving patient motivation in game development for motor deficit rehabilitation”, in ACM international conference on advances in computer entertainment technology (2008), pp. 381–384.
- 38 G. D. Forney Jr, “The Viterbi algorithm”, *Proceedings of the IEEE* **61**, 268–278 (1973).
- 39 J. Fox and J. N. Bailenson, “Virtual self-modeling: the effects of vicarious reinforcement and identification on exercise behaviors”, *Media Psychology* **12**, 1–25 (2009).
- 40 E. Frank, M. Hall, L. Trigg, G. Holmes, and I. H. Witten, “Data mining in bioinformatics using Weka”, *Bioinformatics* **20**, 2479–2481 (2004).
- 41 V. Frati and D. Prattichizzo, “Using Kinect for hand tracking and rendering in wearable haptics”, in IEEE world haptics conference (WHC) (2011), pp. 317–321.
- 42 Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm”, in *Machine learning* (Morgan Kaufmann Publishers, Inc., 1996), pp. 148–156.
- 43 K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: control, sensing, vision, and intelligence* (McGraw-Hill, Inc., 1987).
- 44 H. Fujiyoshi and A. J. Lipton, “Real-time human motion analysis by image skeletonization”, in Fourth workshop on applications of computer vision (wacv’98) (1998), pp. 15–21.

- 45 L. Gallo, A. P. Placitelli, and M. Ciampi, “Controller-free exploration of medical image data: Experiencing the Kinect”, in 24th International Symposium on Computer-Based Medical Systems (CBMS) (2011), pp. 1–6.
- 46 A. Ghodsi, “Dimensionality reduction a short tutorial”, Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada (2006).
- 47 M. Gleicher, “Retargetting motion to new characters”, in Proceedings of the 25th annual conference on computer graphics and interactive techniques (ACM, 1998), pp. 33–42.
- 48 I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: real-time surveillance of people and their activities”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **22**, 809–830 (2000).
- 49 T. Hastie, R. Tibshirani, et al., “Classification by pairwise coupling”, *The annals of statistics* **26**, 451–471 (1998).
- 50 T. Horprasert, D. Harwood, and L. S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection”, in *Ieee iccv*, Vol. 99 (1999), pp. 1–19.
- 51 W. IJsselsteijn, Y. de Kort, J. Westerink, M. de Jager, and R. Bonants, “Fun and sports: enhancing the home fitness experience”, in *Entertainment computing (ICEC)* (Springer, 2004), pp. 46–56.
- 52 K. Ilgun, R. A. Kemmerer, and P. A. Porras, “State transition analysis: a rule-based intrusion detection approach”, *Software Engineering, IEEE Transactions on* **21**, 181–199 (1995).
- 53 O. C. Jenkins and M. J. Mataric, “Deriving action and behavior primitives from human motion data”, in *IEEE/RSJ international conference on intelligent robots and systems*, Vol. 3 (2002), pp. 2551–2556.
- 54 G. H. John and P. Langley, “Estimating continuous distributions in bayesian classifiers”, in *Eleventh conference on uncertainty in artificial intelligence* (1995), pp. 338–345.
- 55 S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to Platt’s SMO algorithm for SVM classifier design”, *Neural Computation* **13**, 637–649 (2001).
- 56 R. Kehl, M. Bray, and L. V. Gool, “Full body tracking from multiple views using stochastic sampling”, in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2 (IEEE, 2005), pp. 129–136.

- 57 K. Khoshelham and S. O. Elberink, “Accuracy and resolution of Kinect depth data for indoor mapping applications”, *Sensors* **12**, 1437–1454 (2012).
- 58 S.-K. Kim, J. Chun, H. Park, D. Kwon, and Y. Ro, “Interfacing sensors and virtual world health avatar application”, in 5th IEEE International Conference on New Trends in Information Science and Service Science (NISS), Vol. 1 (2011), pp. 21–25.
- 59 R. Klaassen, T. Lavrysen, G. Geleijnse, A. van Halteren, H. Schwieter, M. van der Hout, et al., “A personal context-aware multi-device coaching service that supports a healthy lifestyle”, in 25th BCS Conference on Human-Computer Interaction (2011), pp. 443–448.
- 60 D. E. Knuth, J. H. Morris, Jr, and V. R. Pratt, “Fast pattern matching in strings”, *SIAM journal on computing* **6**, 323–350 (1977).
- 61 S. Kumar, “Classification and detection of computer intrusions”, PhD thesis (Purdue University, 1995).
- 62 C. Kuo-Liang, “An improved algorithm for solving the banded cyclic string-to-string correction problem”, *Theoretical Computer Science* **201**, 275–279 (1998).
- 63 G. Kurillo, J. J. Han, A. Nicorici, and R. Bajcsy, “Tele-MFAsT: Kinect-based Tele-Medicine tool for remote motion and function assessment”, *Medicine Meets Virtual Reality 21: NextMed/MMVR21* **196**, 215 (2014).
- 64 E. B. Larson, L. Wang, J. D. Bowen, W. C. McCormick, L. Teri, P. Crane, and W. Kukull, “Exercise is associated with reduced risk for incident dementia among persons 65 years of age and older”, *Annals of internal medicine* **144**, 73–81 (2006).
- 65 J. Lee, P. Karasev, L. Zhu, and A. Tannenbaum, “Human body tracking and joint angle estimation from mobile-phone video for clinical analysis”, in (June 2011).
- 66 J. Lee, J. Chai, P. S. Reitsma, J. K. Hodgins, and N. S. Pollard, “Interactive control of avatars animated with human motion data”, in *ACM Transactions on Graphics (TOG)*, Vol. 21, 3 (ACM, 2002), pp. 491–500.
- 67 J. Lee and S. Y. Shin, “A hierarchical approach to interactive motion editing for human-like figures”, in *Proceedings of the 26th annual conference on computer graphics and interactive techniques (ACM Press/Addison-Wesley Publishing Co., 1999)*, pp. 39–48.
- 68 L. A. Leiva and E. Vidal, “Warped K-Means: An algorithm to cluster sequentially-distributed data”, *Information Sciences* **237**, 196–210 (2013).

- 69 V. Lepetit and P. Fua, “Monocular model-based 3D tracking of rigid objects: A survey”, *Foundations and trends in computer graphics and vision* **1**, 1–89 (2005).
- 70 H. Levene, “Ingram olkin, harold hotelling, et alia, ed”, *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, 278–292 (1960).
- 71 Y. Li, C. Fermuller, Y. Aloimonos, and H. Ji, “Learning shift-invariant sparse representation of actions”, in *Conference on computer vision and pattern recognition (CVPR)* (2010), pp. 2630–2637.
- 72 C. Lisetti, F. Nasoz, C. LeRouge, O. Ozyer, and K. Alvarez, “Developing multimodal intelligent affective interfaces for tele-home health care”, *International Journal of Human-Computer Studies* **59**, 245–255 (2003).
- 73 M. Maes, “On a cyclic string-to-string correction problem”, *Information Processing Letters* **35**, 73–78 (1990).
- 74 D. Maier, “The complexity of some problems on subsequences and supersequences”, *Journal of the ACM (JACM)* **25**, 322–336 (1978).
- 75 M. Meredith and S. Maddock, “Motion capture file formats explained”, Department of Computer Science, University of Sheffield, Technical report CS-01-11 (2001).
- 76 M. J. Meredith, “Adapting and reconfiguring human figure motion capture data through the application of inverse kinematics and biomechanics-based optimisation”, PhD thesis (University of Sheffield, 2005).
- 77 D. I. Miller, V. Taler, P. S. Davidson, and C. Messier, “Measuring the impact of exercise on cognitive aging: methodological issues”, *Neurobiology of aging* **33**, 622–629 (2012).
- 78 M. Mitchell, S. Forrest, and J. H. Holland, “The royal road for genetic algorithms: fitness landscapes and ga performance”, in *Proceedings of the first european conference on artificial life* (Cambridge: The MIT Press, 1992), pp. 245–254.
- 79 A. Möller, L. Roalter, S. Diewald, J. Scherr, M. Kranz, N. Hammerla, P. Olivier, and T. Plötz, “Gymskill: A personal trainer for physical exercises”, in *IEEE international conference on pervasive computing and communications (percom)* (2012), pp. 213–220.
- 80 L. S. Moualed, V. M. Jones, H. J. Hermens, et al., “A self-learning personalized feedback agent for motivating physical activity”, in *4th ACM international symposium on applied sciences in biomedical and communication technologies* (2011), p. 147.

- 81 D. W. Mount, “Sequence and genome analysis”, Bioinformatics: Cold Spring Harbour Laboratory Press: Cold Spring Harbour **2** (2004).
- 82 M. Müller and T. Röder, “Motion templates for automatic classification and retrieval of motion capture data”, in Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer Animation (Eurographics Association, 2006), pp. 137–146.
- 83 S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, *Journal of molecular biology* **48**, 443–453 (1970).
- 84 S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, “Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population”, in Annual international conference of the Engineering in medicine and biology society (EMBC) (IEEE, 2012), pp. 1188–1193.
- 85 Š. Obdržálek, G. Kurillo, E. Seto, and R. Bajcsy, “Architecture of an automated coaching system for elderly population”, *Studies in health technology and informatics* **184**, 309 (2013).
- 86 F. Ofli, G. Kurillo, S. Obdrzalek, R. Bajcsy, H. Jimison, and M. Pavel, “Design and evaluation of an interactive exercise coaching system for older adults: lessons learned”, (2015).
- 87 F. J. Ordóñez, P. de Toledo, and A. Sanchis, “Activity recognition using hybrid generative/discriminative models on home environments using binary sensors”, *Sensors* **13**, 5460–5477 (2013).
- 88 T. J. Parr and R. W. Quong, “Antlr: a predicated-ll (k) parser generator”, *Software: Practice and Experience* **25**, 789–810 (1995).
- 89 M. Pirovano, R. Mainetti, G. Baud-Bovy, P. L. Lanzi, and N. A. Borghese, “Self-adaptive games for rehabilitation at home”, in IEEE conference on computational intelligence and games (cig) (2012), pp. 179–186.
- 90 N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, “Adapting human motion for the control of a humanoid robot”, in International Conference on Robotics and Automation (ICRA’02), Vol. 2 (IEEE, 2002), pp. 1390–1397.
- 91 PrimeSense, *Primesense natural interaction*, <http://www.primesense.com/>, Accessed: 8/13/2012, 2012.

- 92 D. S. PT, *Eldergym*, <http://www.eldergym.com/exercises.html>, Accessed: 2015, 2015.
- 93 J. R. Quinlan, *C4. 5: programs for machine learning*, Vol. 1 (Morgan kaufmann, 1993).
- 94 M. Raptis, D. Kirovski, and H. Hoppe, “Real-time classification of dance gestures from skeleton animation”, in *ACM siggraph/eurographics symposium on computer animation* (ACM, 2011), pp. 147–156.
- 95 Z. Ren, J. Meng, J. Yuan, and Z. Zhang, “Robust hand gesture recognition with kinect sensor”, in *Proceedings of the 19th ACM international conference on multimedia* (2011), pp. 759–760.
- 96 Y. Rui and P. Anandan, “Segmenting visual actions based on spatio-temporal motion patterns”, in *Conference on computer vision and pattern recognition*, Vol. 1 (2000), pp. 111–118.
- 97 Z. Ruttkay and H. van Welbergen, “Elbows higher! performing, observing and correcting exercises by a virtual trainer”, in *Intelligent virtual agents* (Springer, 2008), pp. 409–416.
- 98 K. Sakurai, W. Choi, L. Li, and K. Hachimura, “Retrieval of similar behavior data using kinect data”, in *Control, automation and systems (iccas), 2014 14th international conference on* (IEEE, 2014), pp. 1368–1372.
- 99 S. Salvador and P. Chan, “Toward accurate dynamic time warping in linear time and space”, *Intelligent Data Analysis* **11**, 561–580 (2007).
- 100 W. M. Sauter, “A comparison of Wii(tm) exergaming and matter of balance on aspects of balance and activity adherence in older adults”, MA thesis (East Carolina University, 2011).
- 101 L. A. Schwarz, A. Mkhitarian, D. Mateus, and N. Navab, “Human skeleton tracking from depth data using geodesic distances and optical flow”, *Image and Vision Computing* **30**, 217–226 (2012).
- 102 L. Shao, L. Ji, Y. Liu, and J. Zhang, “Human action segmentation and recognition via motion and shape analysis”, *Pattern Recognition Letters* **33**, 438–445 (2012).
- 103 R. Sharma, M. A. Alam, and A. Rani, “K-Means clustering in spatial data mining using Weka interface”, *International Conference on Advances in Communication and Computing Technologies ICACACT*, 26–30 (2012).

- 104 T. Shiratori, A. Nakazawa, and K. Ikeuchi, “Rhythmic motion analysis using motion capture and musical information”, in Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2003) (2003), pp. 89–94.
- 105 J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, “Real-time human pose recognition in parts from single depth images”, *Communications of the ACM* **56**, 116–124 (2013).
- 106 J. Smisek, M. Jancosek, and T. Pajdla, “3D with Kinect”, in *Consumer depth cameras for computer vision* (Springer, 2013), pp. 3–25.
- 107 T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences”, *Journal of molecular biology* **147**, 195–197 (1981).
- 108 G. Snedecor, *Statistical methods: by george w. snedecor and william g. cochrane* (Iowa State University Press, 1989).
- 109 D. Spelmezan, A. Schanowski, and J. Borchers, “Wearable automatic feedback devices for physical activities”, in Fourth international conference on body area networks (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009), p. 1.
- 110 J. Stowers, M. Hayes, and A. Bainbridge-Smith, “Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor”, in IEEE international conference on mechatronics (ICM) (2011), pp. 358–362.
- 111 H. Takahashi, N. Itoh, T. Amano, and A. Yamashita, “A spelling correction method and its application to an OCR system”, *Pattern Recognition* **23**, 363–377 (1990).
- 112 J. K. T. Tang, J. C. P. Chan, and H. Leung, “Interactive dancing game with real-time recognition of continuous dance moves from 3D human motion capture”, in Proceedings of the 5th international conference on ubiquitous information management and communication (2011), p. 50.
- 113 K. T. Tang, H. Leung, T. Komura, and H. P. H. Shum, “Finding repetitive patterns in 3D human motion captured data”, in Proceedings of the 2nd international conference on ubiquitous information management and communication (2008), pp. 396–403.
- 114 H. Tange, S. van Genderen, S. van der Weegen, A. Moser, and G. Plasqui, “A pilot with exergames in elderly homes”, in 23rd international conference of the european federation for medical informatics (2011).

- 115 M. J. Taylor, D. McCormick, R. Impson, T. Shawis, and M. Griffin, “Activity promoting gaming systems in exercise and rehabilitation.”, *Journal of Rehabilitation Research and Development* **48**, 1171–1186 (2011).
- 116 A. Tiskin, “Periodic string comparison”, in *Combinatorial pattern matching* (Springer, 2009), pp. 193–206.
- 117 M. Tölgyessy and P. Hubinsky, “The Kinect sensor in robotics education”, in *Proceedings of 2nd international conference on robotics in education* (2011), pp. 143–146.
- 118 T. Van Kasteren, G. Englebienne, and B. J. Kröse, “An activity monitoring system for elderly care using generative and discriminative models”, *Personal and ubiquitous computing* **14**, 489–498 (2010).
- 119 J. Verghese, R. B. Lipton, M. J. Katz, C. B. Hall, C. A. Derby, G. Kuslansky, A. F. Ambrose, M. Sliwinski, and H. Buschke, “Leisure activities and the risk of dementia in the elderly”, *New England Journal of Medicine* **348**, 2508–2516 (2003).
- 120 ViconSystems, *Vicon systems - vicon bonita*, <http://www.vicon.com/products/bonita.html>, Accessed: 8/13/2012, 2012.
- 121 G. I. Webb, “Decision tree grafting from the all-tests-but-one partition”, in *16th International Joint Conference on Artificial Intelligence (IJCAI’99)*, Vol. 2 (1999), pp. 702–707.
- 122 D. Weinland, R. Ronfard, and E. Boyer, “A survey of vision-based methods for action representation, segmentation and recognition”, *Computer Vision and Image Understanding* **115**, 224–241 (2011).
- 123 W. Wieringa, H. O. D. Akker, V. M. Jones, R. O. D. Akker, and H. J. Hermens, “Ontology-based generation of dynamic feedback on physical activity”, in *Artificial intelligence in medicine* (Springer, 2011), pp. 55–59.
- 124 R. Q. Wolever, D. M. Webber, J. P. Meunier, J. M. Greeson, E. R. Lausier, and T. W. Gaudet, “Modifiable disease risk, readiness to change, and psychosocial functioning improve with integrative medicine immersion model”, *Alternative Therapies in Health and Medicine* **17**, 38 (2011).
- 125 L. Xia, C.-C. Chen, and J. Aggarwal, “Human detection using depth information by Kinect”, in *IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW)* (2011), pp. 15–22.

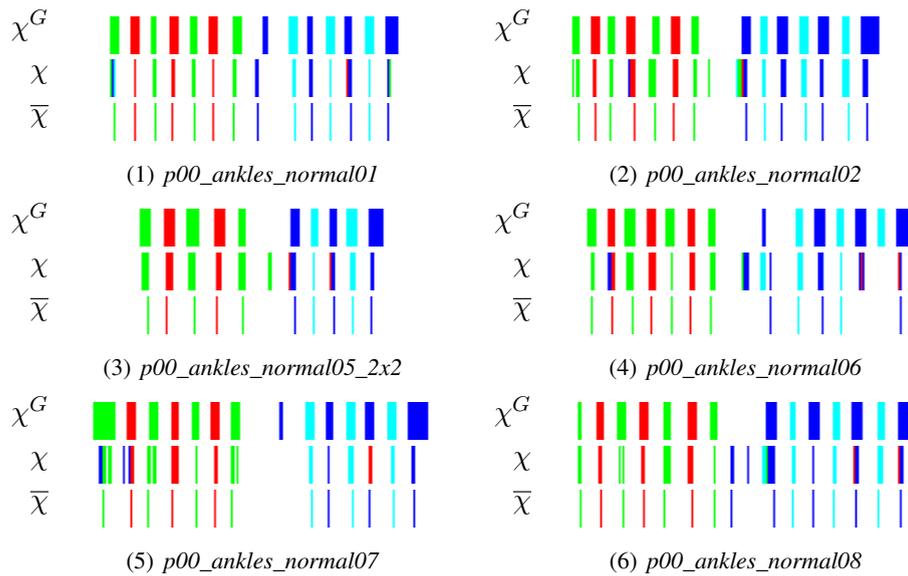
- 
- 126 M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, “Accurate 3d pose estimation from a single depth image”, in *Computer vision (iccv), 2011 ieee international conference on (IEEE, 2011)*, pp. 731–738.
  - 127 F. Zhou, F. De la Torre, and J. K. Hodgins, “Hierarchical aligned cluster analysis for temporal clustering of human motion”, *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)* **35**, 582–596 (2013).

# Appendix A

## Results for landmark identification and landmark refining

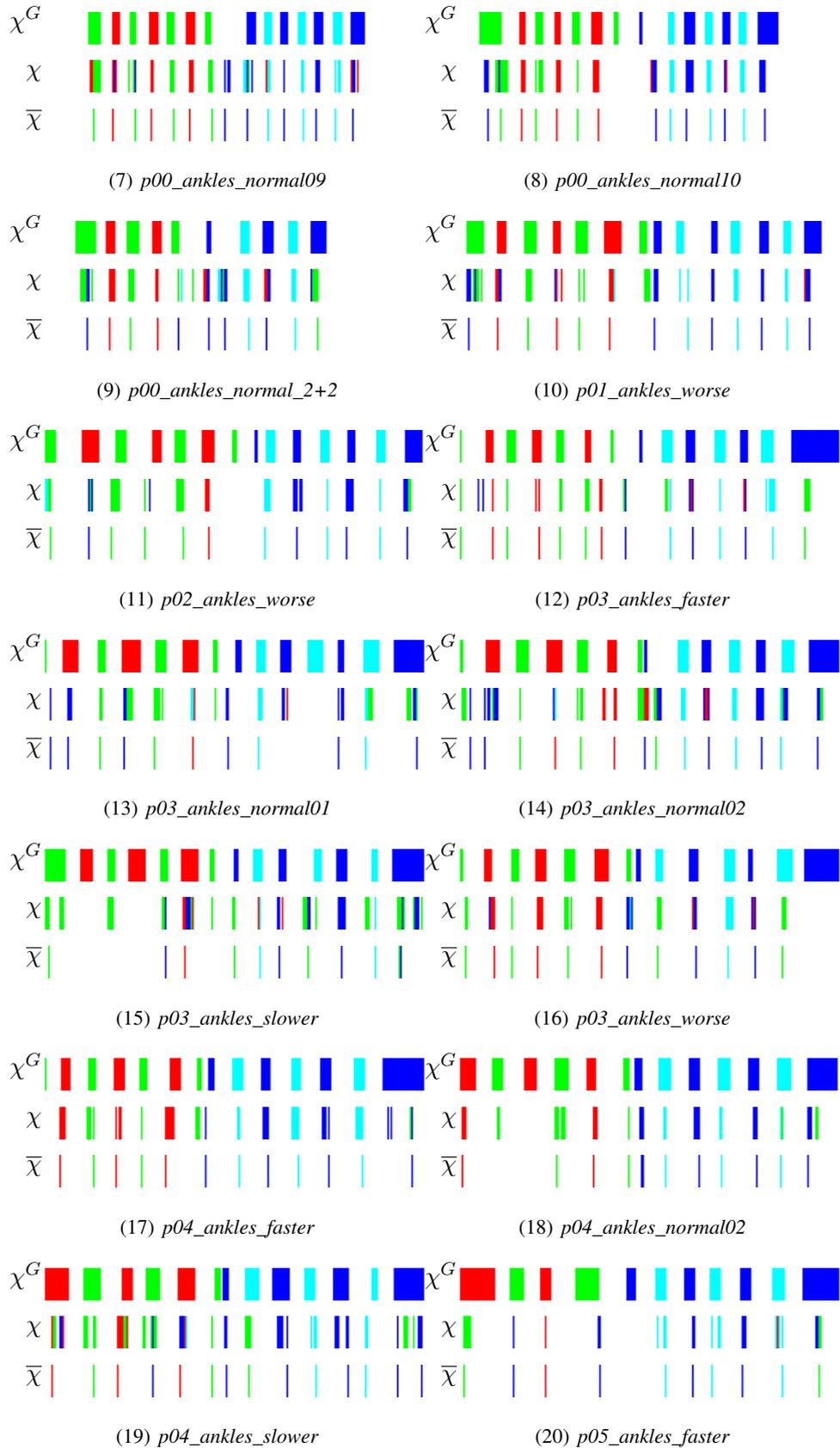
Representation of results for landmark identification and landmark refining. Ground truth landmarks ( $\chi^G$ ), candidate landmarks ( $\chi$ ) and refined landmarks ( $\bar{\chi}$ ) represented with vertical markers of different colours, each of which representing a ground-truth, identified or refined landmark type, respectively.

FIGURE A.1: Representation of results for landmark identification and landmark refining in the *Ankles* exercise



Continued on next page

FIGURE A.1: Representation of results for landmark identification and landmark refining in the *Ankles* exercise (continued from previous page)



Continued on next page

FIGURE A.1: Representation of results for landmark identification and landmark refining in the *Ankles* exercise (continued from previous page)

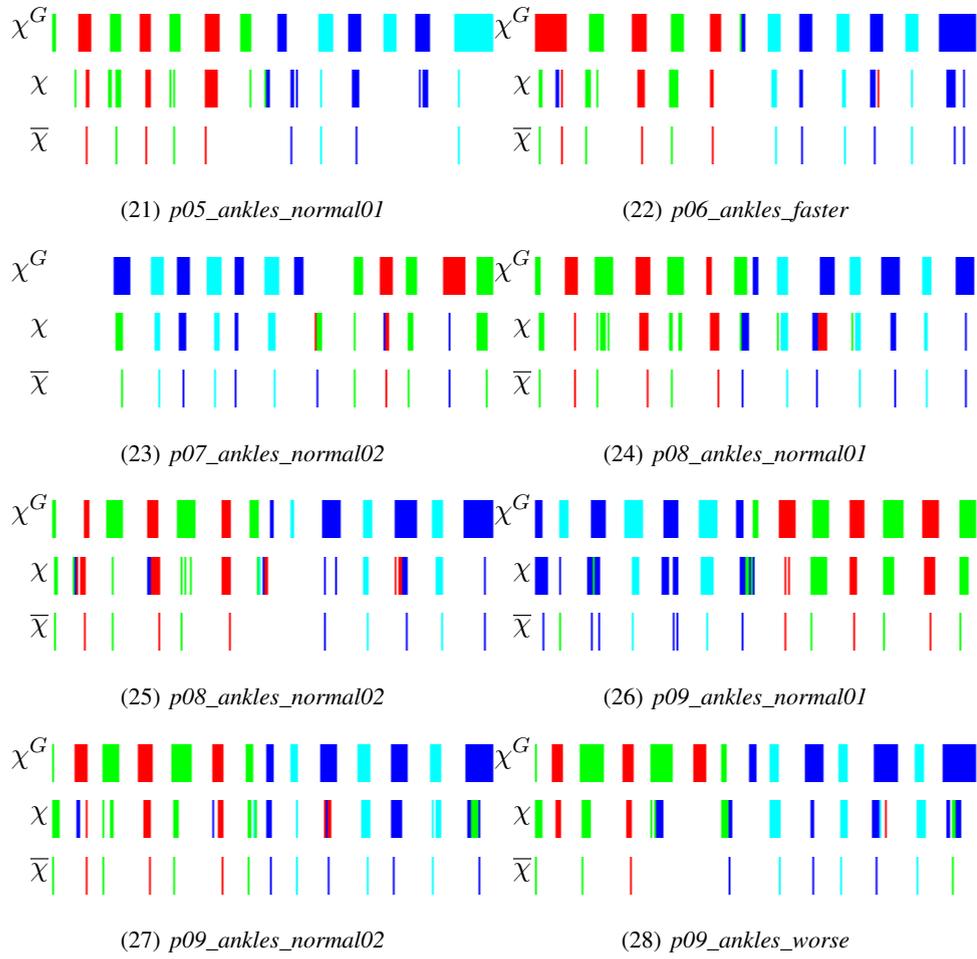
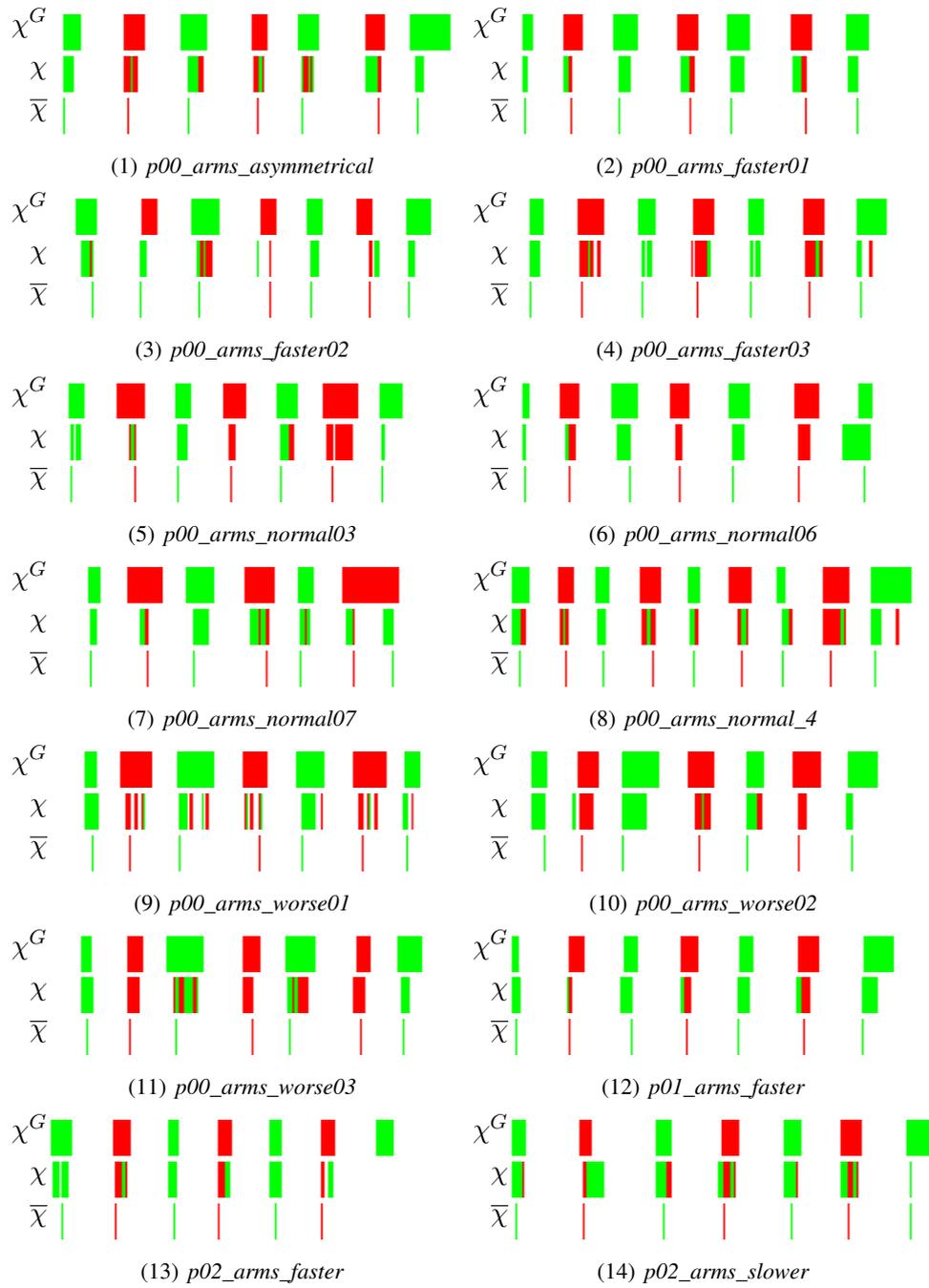
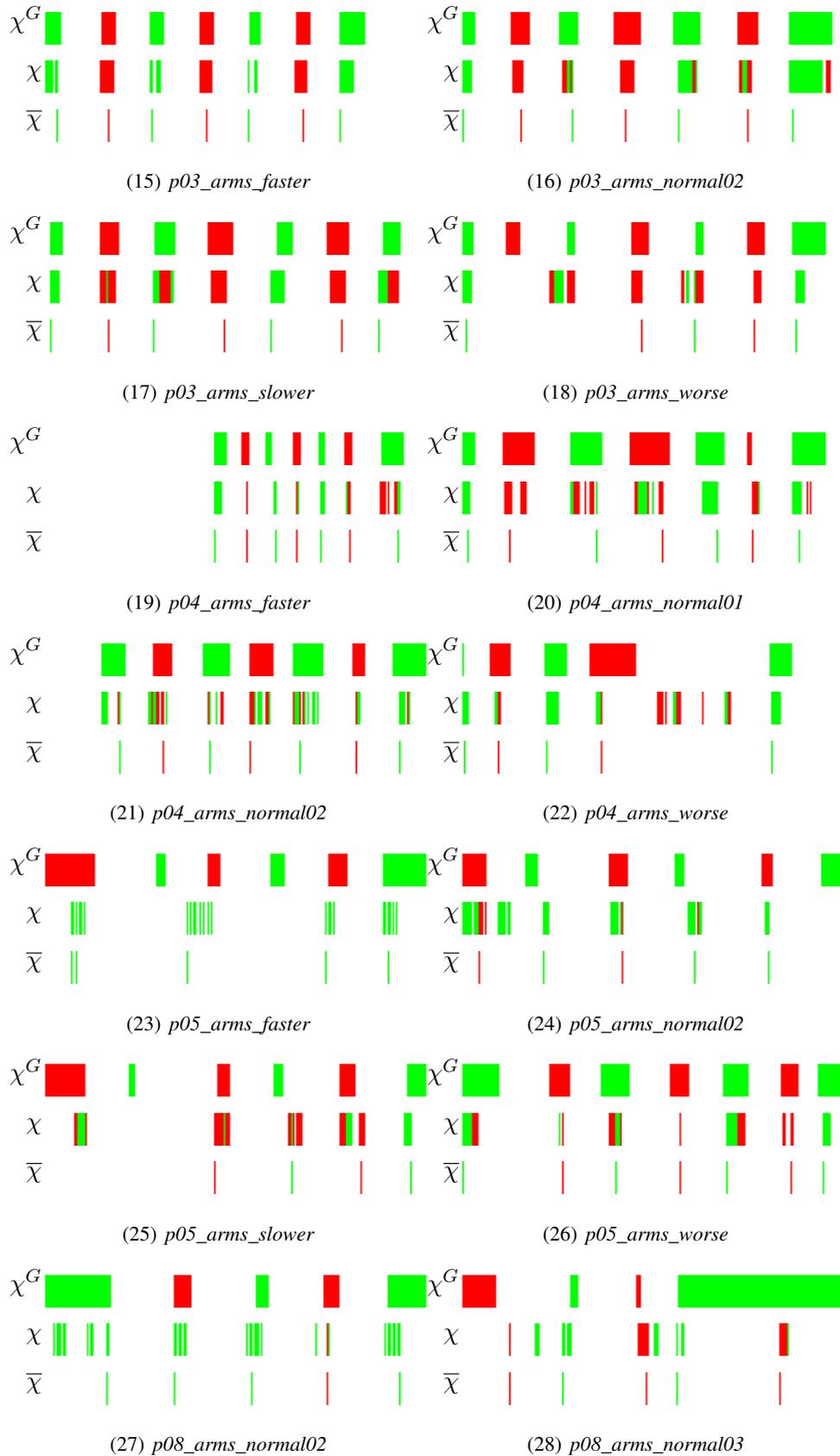


FIGURE A.2: Representation of results for landmark identification and landmark refining in the Arms exercise



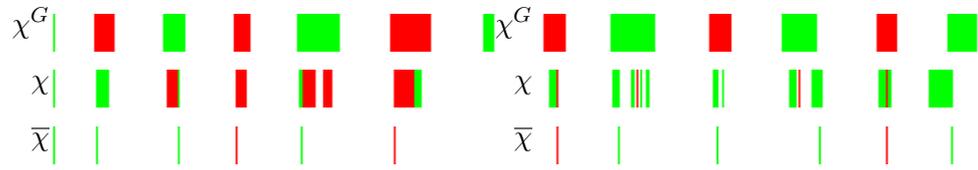
Continued on next page

FIGURE A.2: Representation of results for landmark identification and landmark refining in the Arms exercise (continued from previous page)



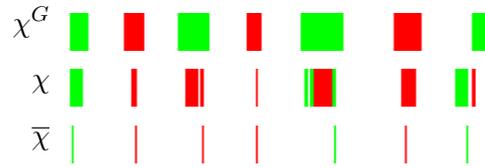
Continued on next page

FIGURE A.2: Representation of results for landmark identification and landmark refining in the Arms exercise (continued from previous page)



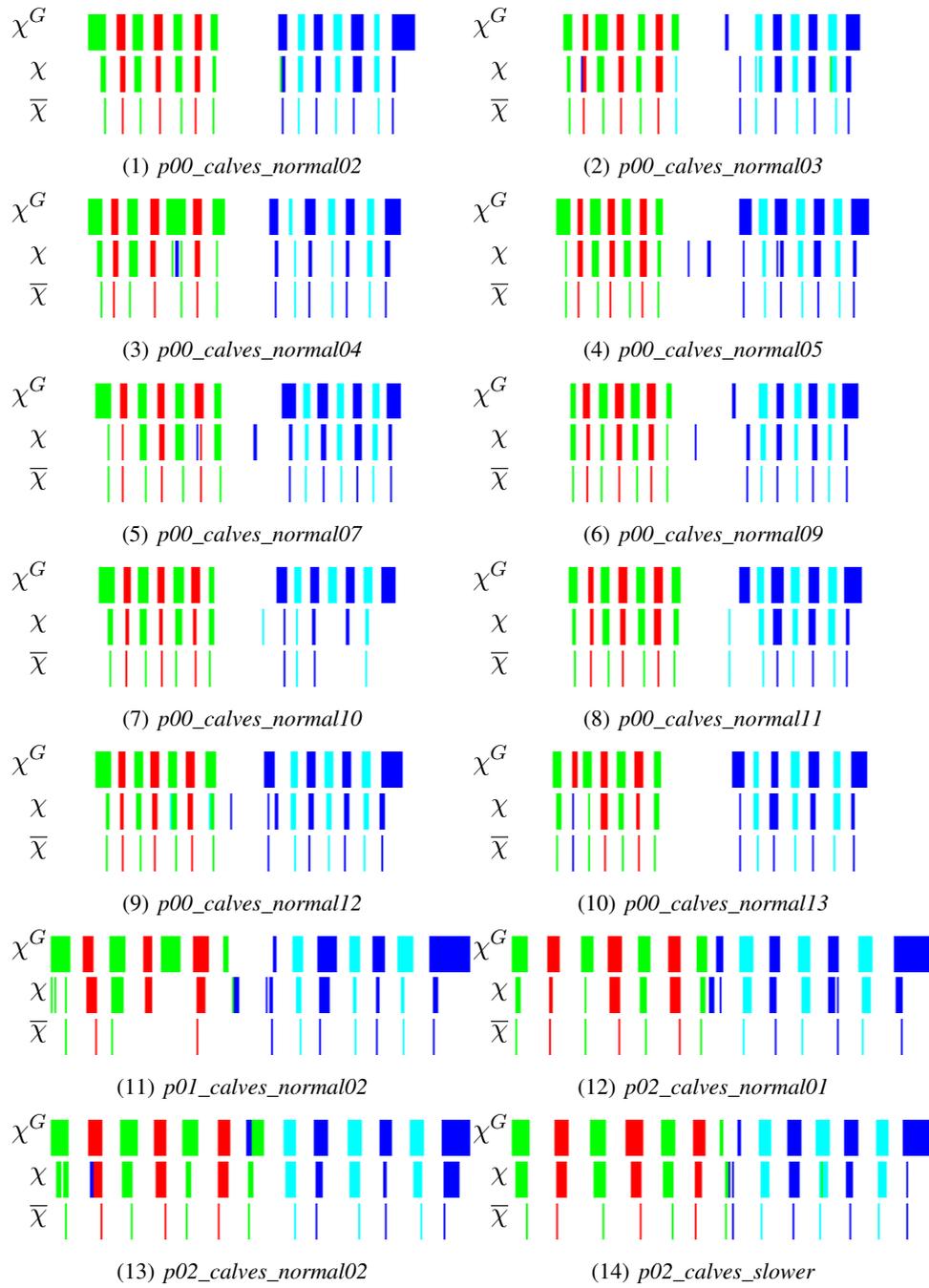
(29) *p09\_arms\_normal01*

(30) *p09\_arms\_normal02*



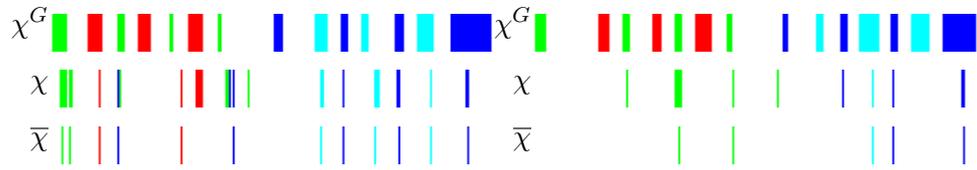
(31) *p09\_arms\_worse*

FIGURE A.3: Representation of results for landmark identification and landmark refining in the *Calves* exercise



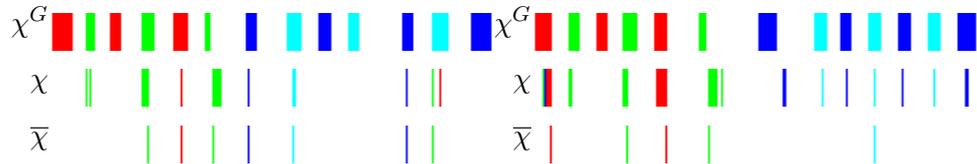
Continued on next page

FIGURE A.3: Representation of results for landmark identification and landmark refining in the *Calves* exercise (continued from previous page)



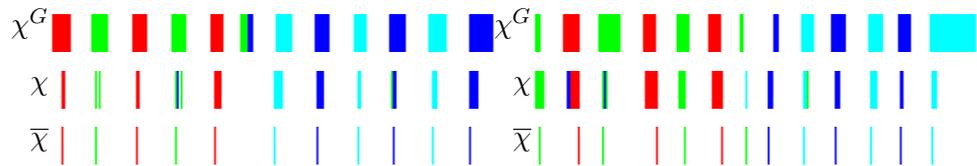
(15) *p03\_calves\_normal02*

(16) *p03\_calves\_slower*



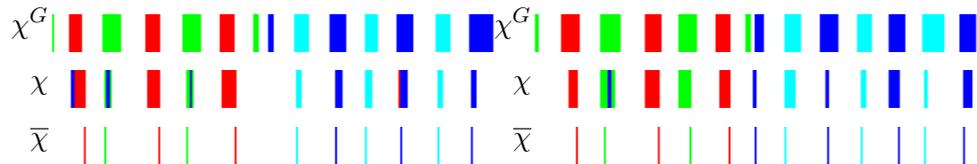
(17) *p05\_calves\_faster*

(18) *p05\_calves\_normal01*



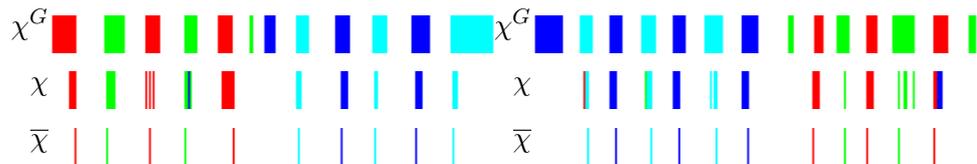
(19) *p06\_calves\_faster*

(20) *p06\_calves\_normal01*



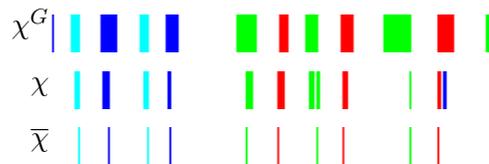
(21) *p06\_calves\_normal02*

(22) *p06\_calves\_slower*



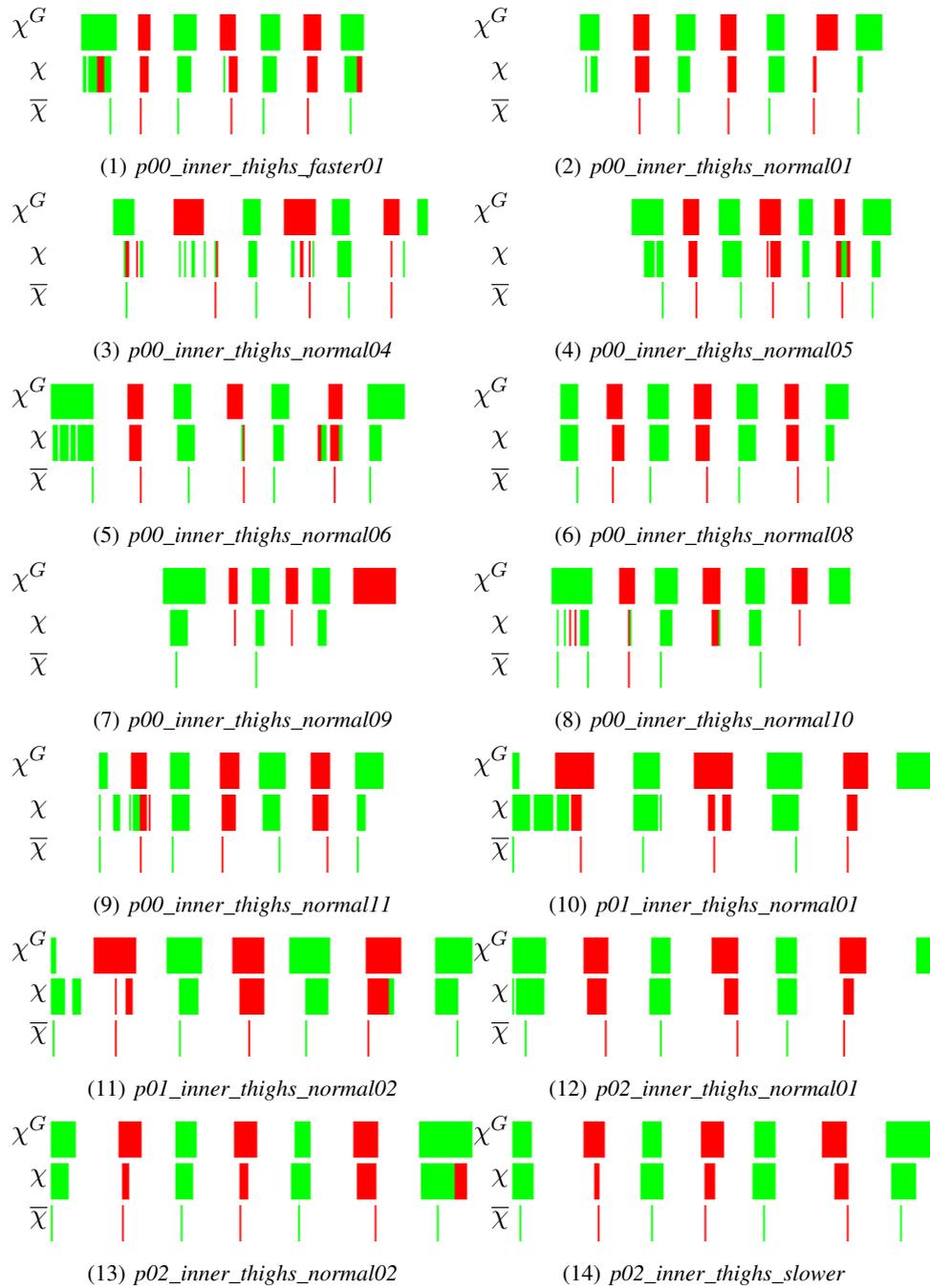
(23) *p06\_calves\_worse*

(24) *p09\_calves\_normal*



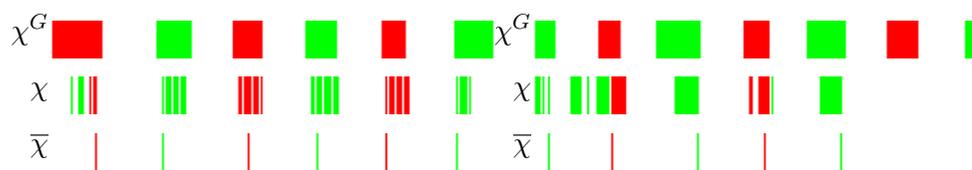
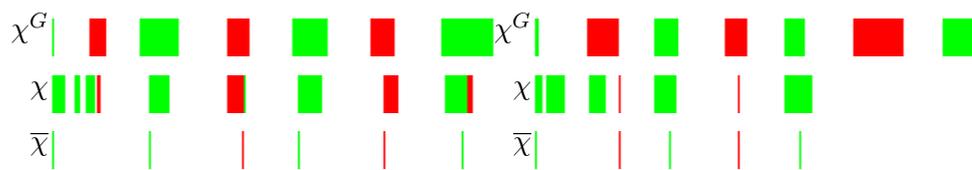
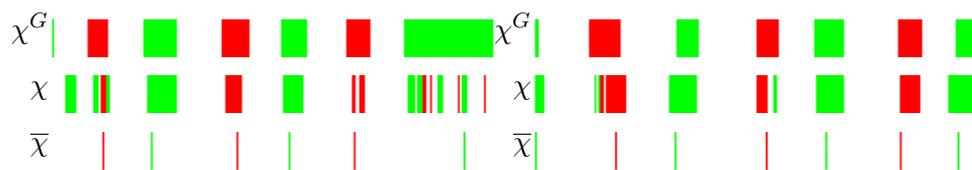
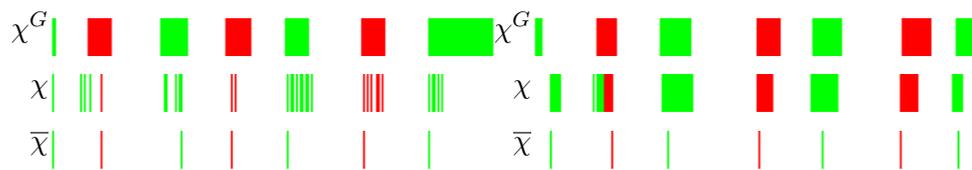
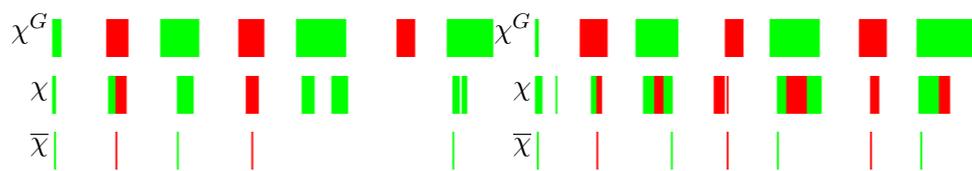
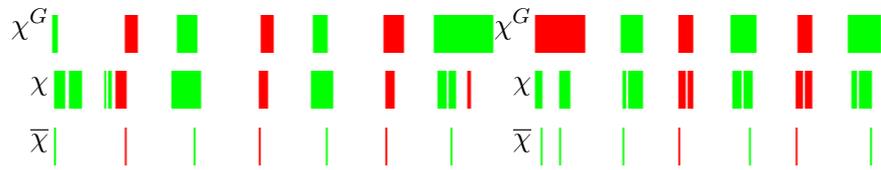
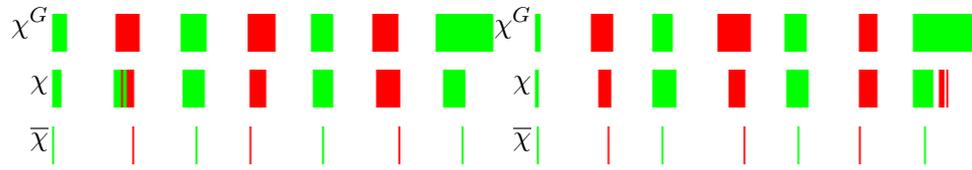
(25) *p09\_calves\_worse*

FIGURE A.4: Representation of results for landmark identification and landmark refining in the *Inner thighs* exercise



Continued on next page

FIGURE A.4: Representation of results for landmark identification and landmark refining in the *Inner thighs* exercise (continued from previous page)



Continued on next page

FIGURE A.4: Representation of results for landmark identification and landmark refining in the *Inner thighs* exercise (continued from previous page)

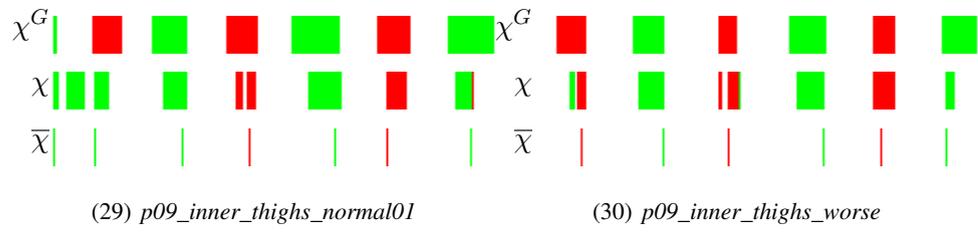
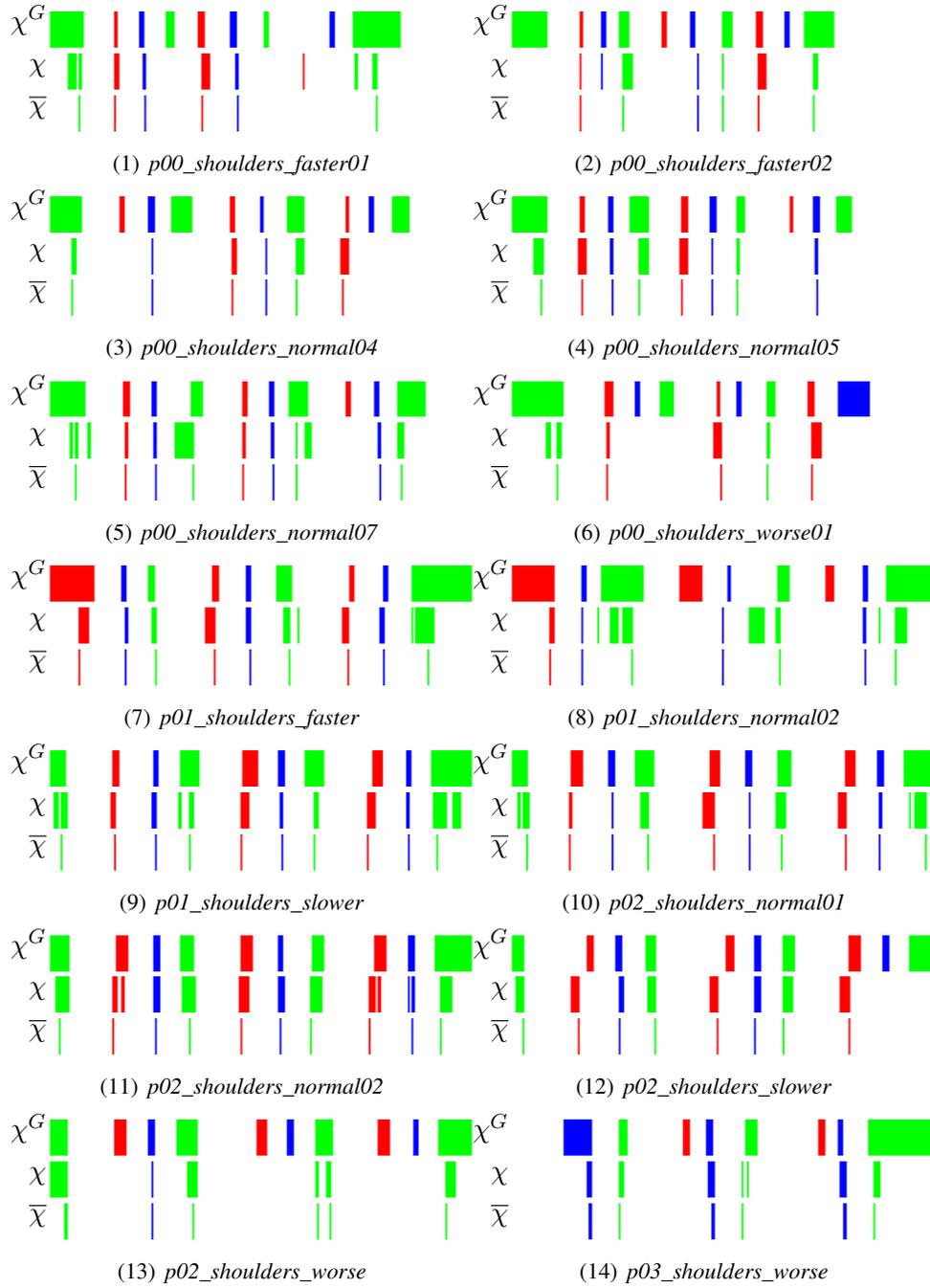
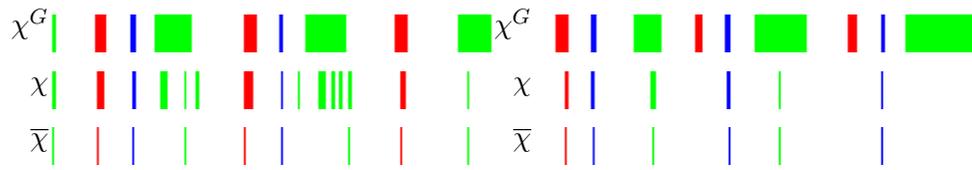
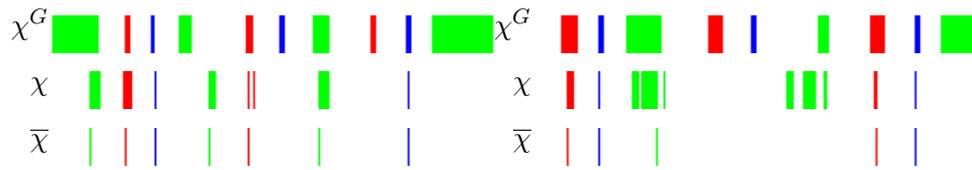
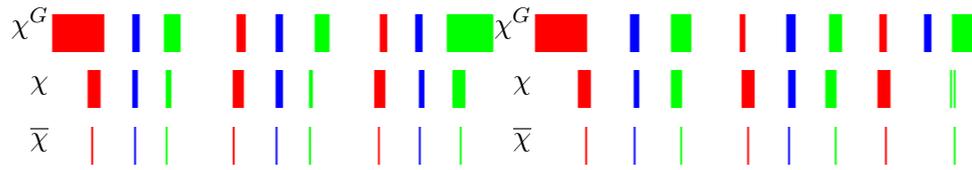
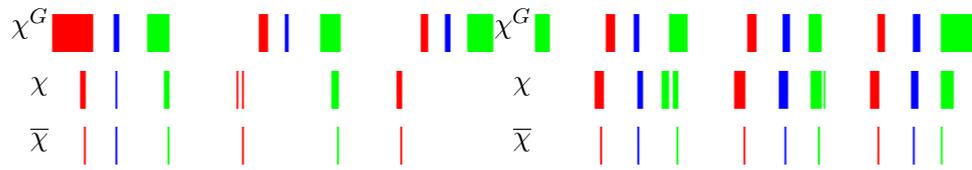
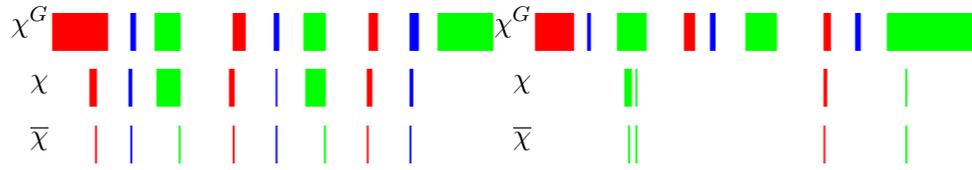
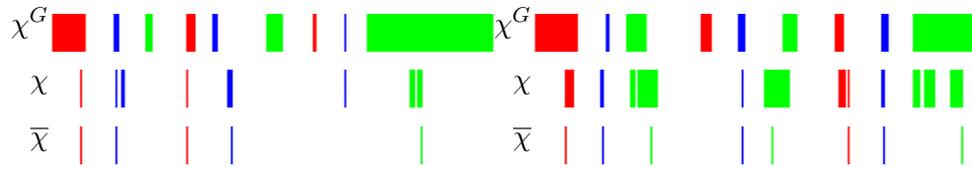


FIGURE A.5: Representation of results for landmark identification and landmark refining in the *Shoulders* exercise



Continued on next page

FIGURE A.5: Representation of results for landmark identification and landmark refining in the *Shoulders* exercise (continued from previous page)

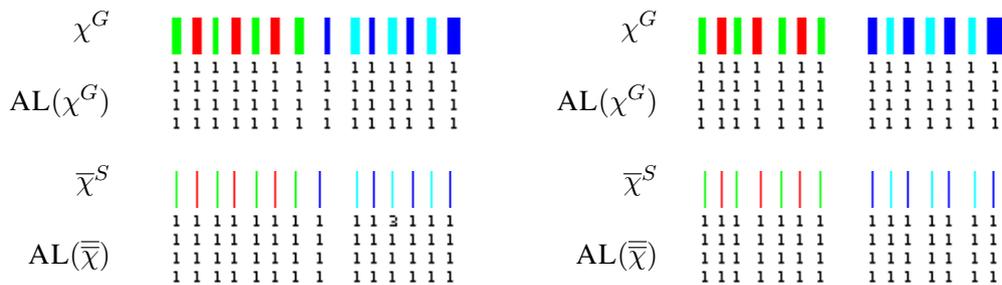


## Appendix B

# Results for periodic landmark sequence analysis and landmark assessment

Representation of results for periodic landmark sequence analysis and landmark assessment. Ground truth landmarks ( $\chi^G$ ) and periodic landmark sequence analysis ( $\bar{\chi}^S$ ) represented with vertical markers of different colours, each of which represents a ground-truth, identified or assessed landmark type, respectively, and their assessment levels ( $AL(\chi^G)$  and  $AL(\bar{\chi})$  respectively), each number standing for an assessment level inferred on an involved body joint, in vertical order (from top to bottom) as per Table 5.2.

FIGURE B.1: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Ankles* exercise

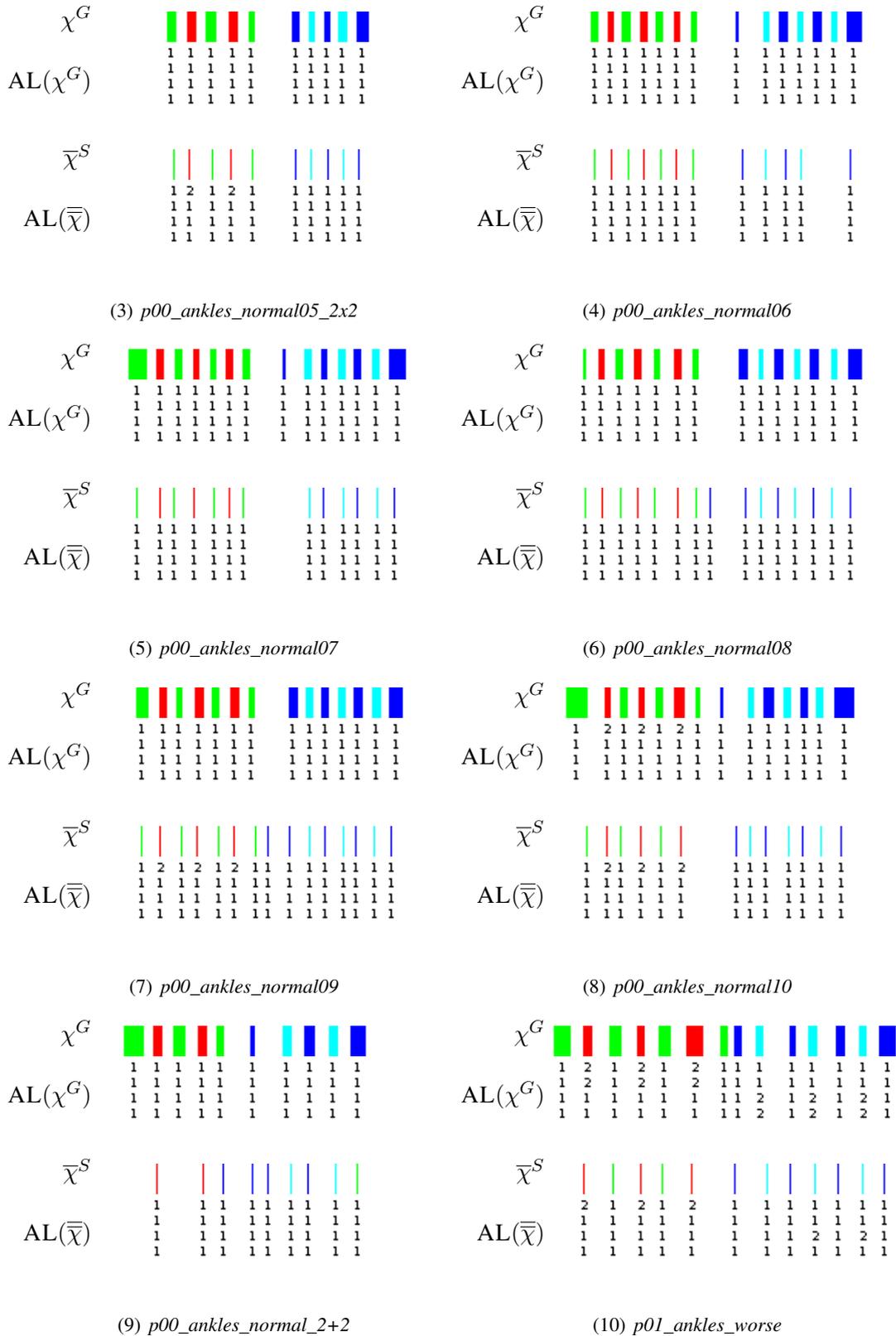


(1) p00\_ankles\_normal01

(2) p00\_ankles\_normal02

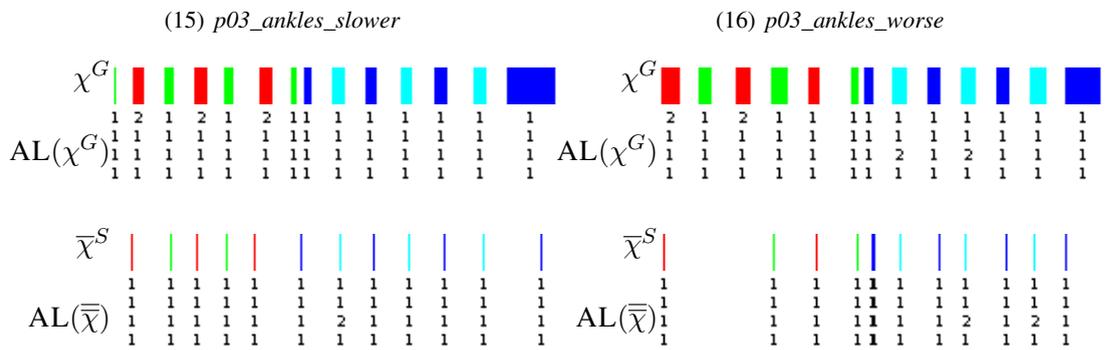
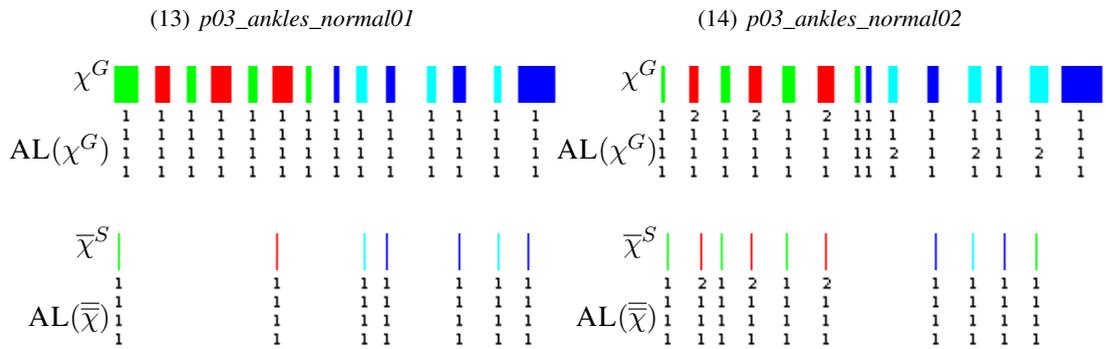
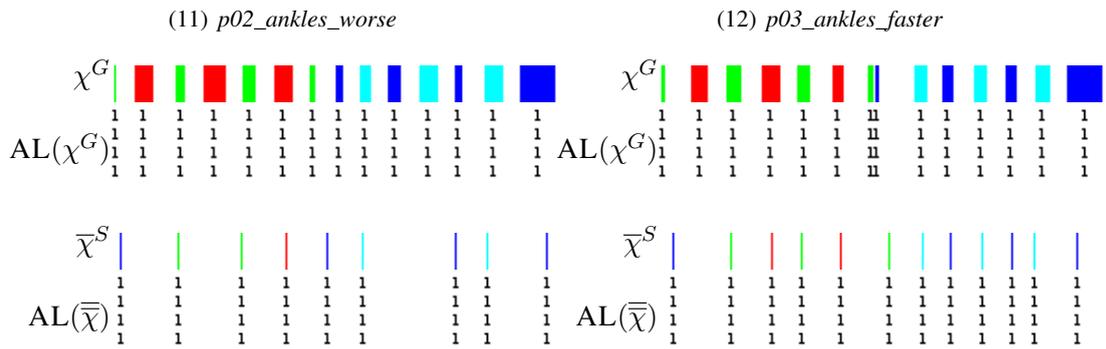
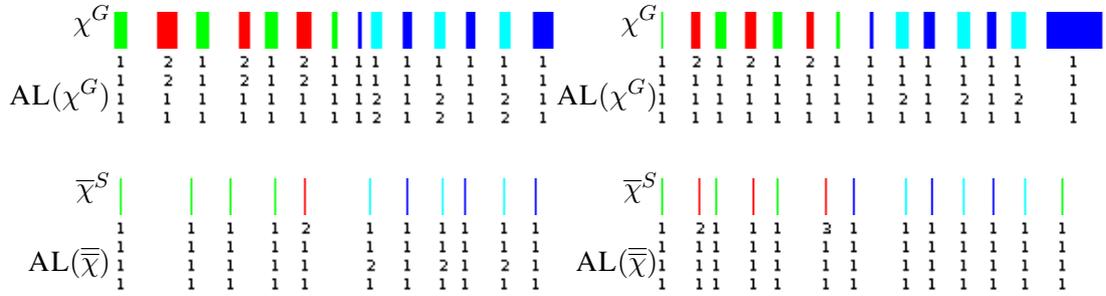
Continued on next page

FIGURE B.1: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Ankles* exercise (continued from previous page)



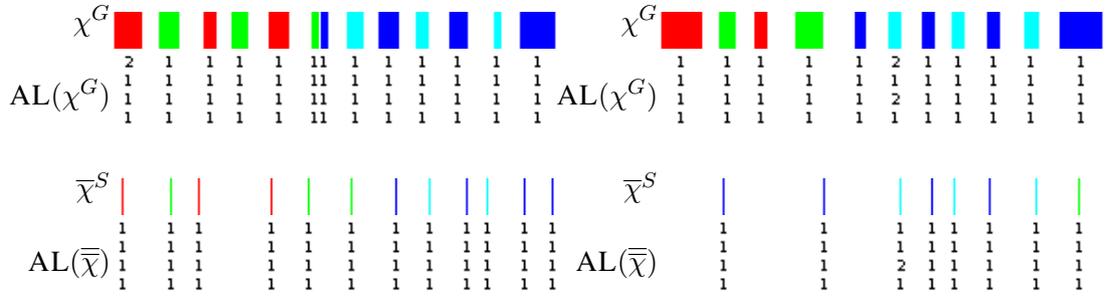
Continued on next page

FIGURE B.1: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Ankles* exercise (continued from previous page)



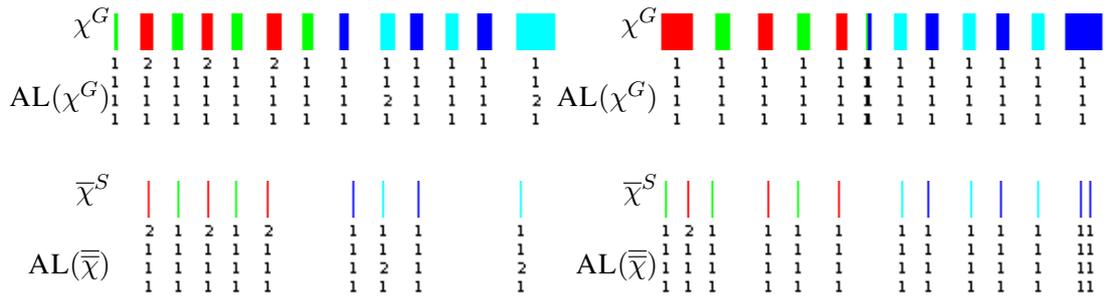
Continued on next page

FIGURE B.1: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Ankles* exercise (continued from previous page)



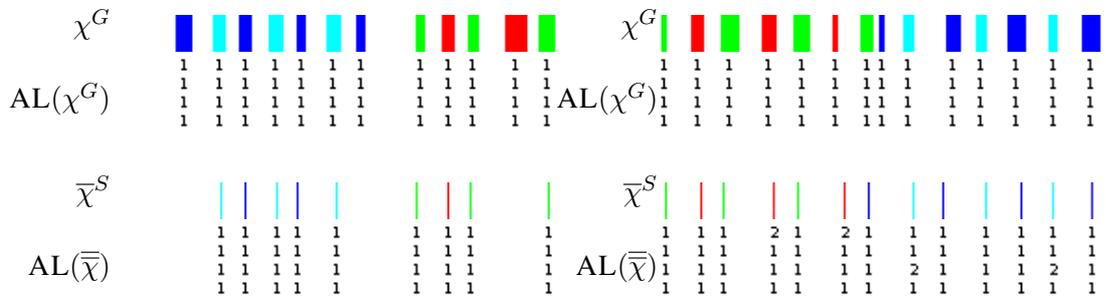
(19) *p04\_ankles\_slower*

(20) *p05\_ankles\_faster*



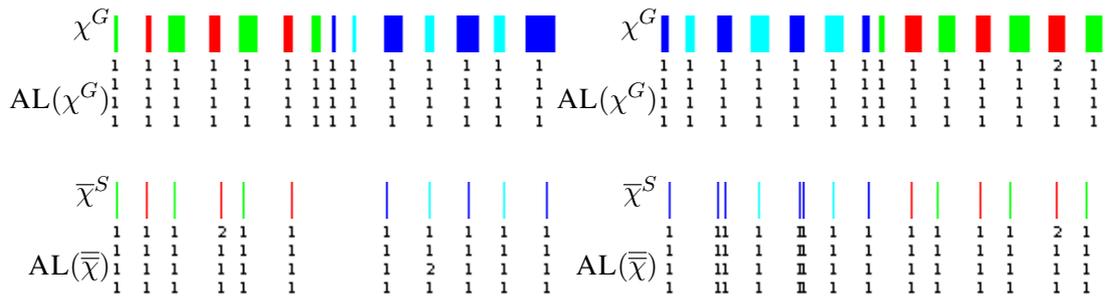
(21) *p05\_ankles\_normal01*

(22) *p06\_ankles\_faster*



(23) *p07\_ankles\_normal02*

(24) *p08\_ankles\_normal01*

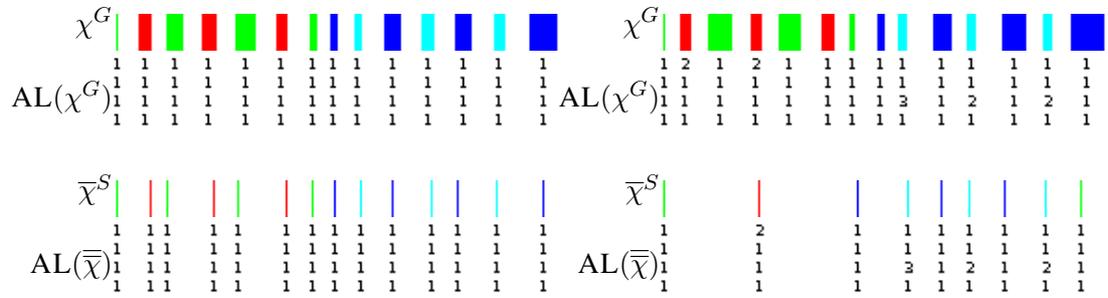


(25) *p08\_ankles\_normal02*

(26) *p09\_ankles\_normal01*

Continued on next page

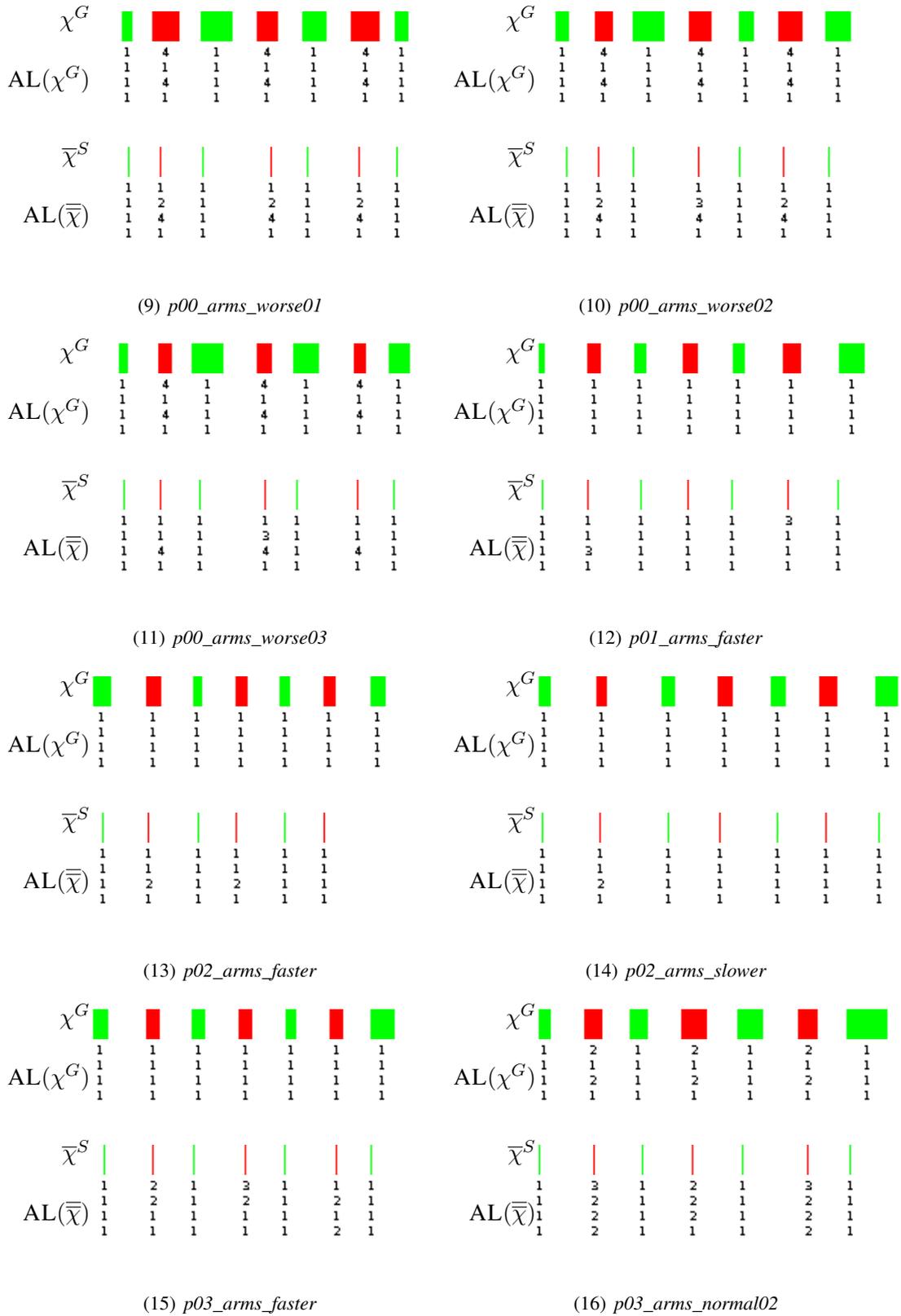
FIGURE B.1: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Ankles* exercise (continued from previous page)



(27) *p09\_ankles\_normal02*

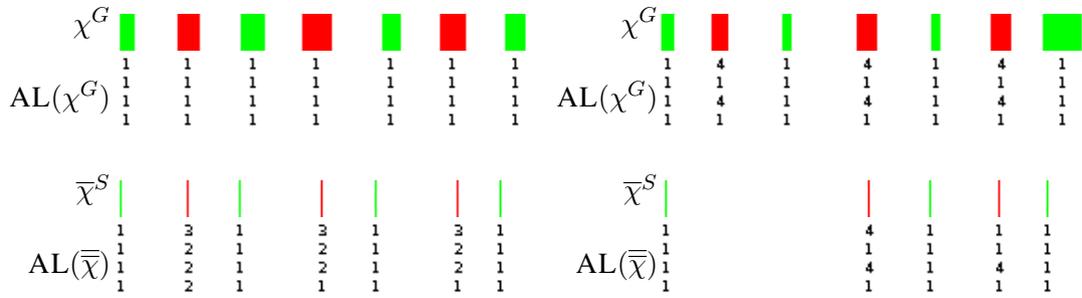
(28) *p09\_ankles\_worse*

FIGURE B.2: Representation of results for periodic landmark sequence analysis and landmark assessment in the Arms exercise (continued from previous page)



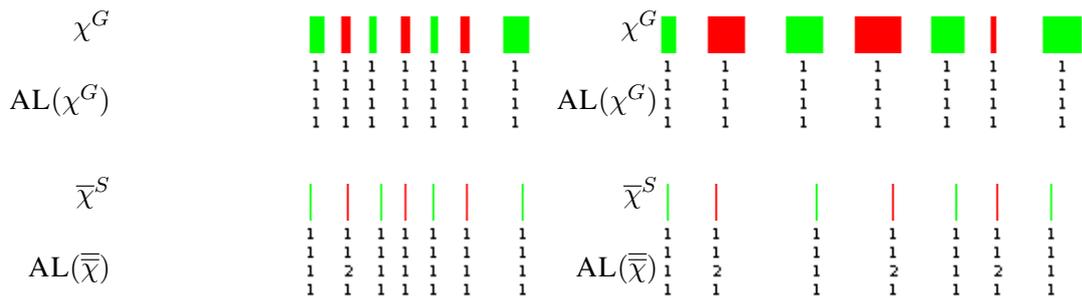
Continued on next page

FIGURE B.2: Representation of results for periodic landmark sequence analysis and landmark assessment in the Arms exercise (continued from previous page)



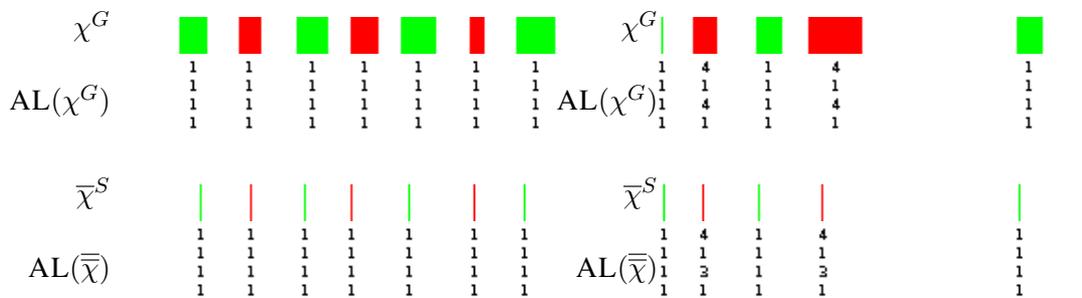
(17) p03\_arms\_slower

(18) p03\_arms\_worse



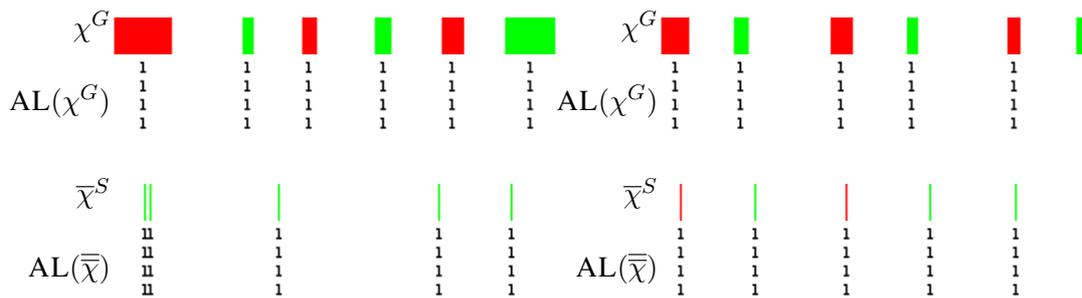
(19) p04\_arms\_faster

(20) p04\_arms\_normal01



(21) p04\_arms\_normal02

(22) p04\_arms\_worse

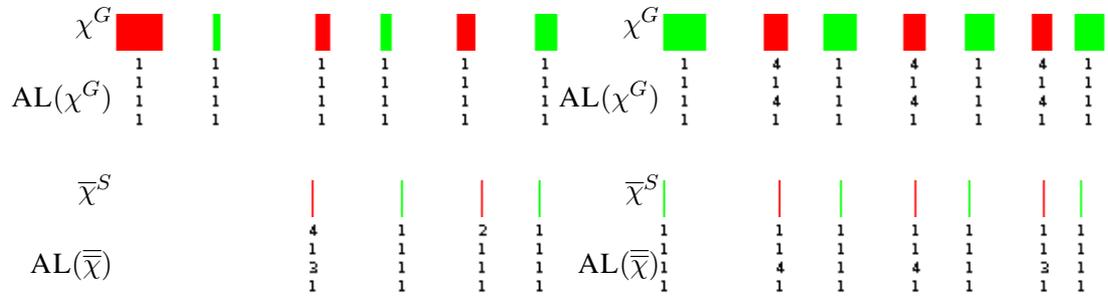


(23) p05\_arms\_faster

(24) p05\_arms\_normal02

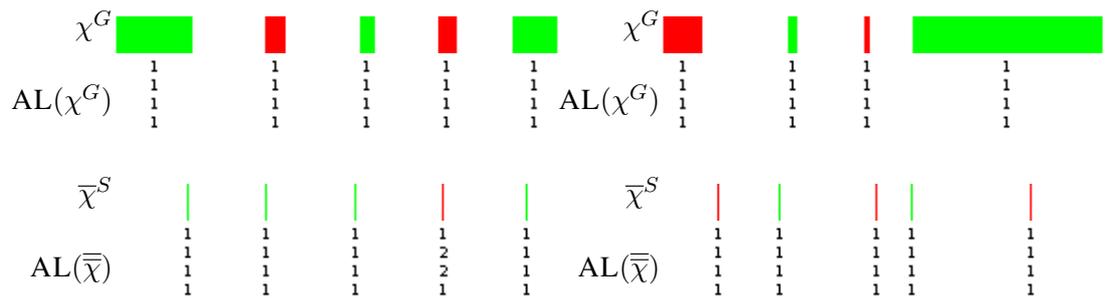
Continued on next page

FIGURE B.2: Representation of results for periodic landmark sequence analysis and landmark assessment in the Arms exercise (continued from previous page)



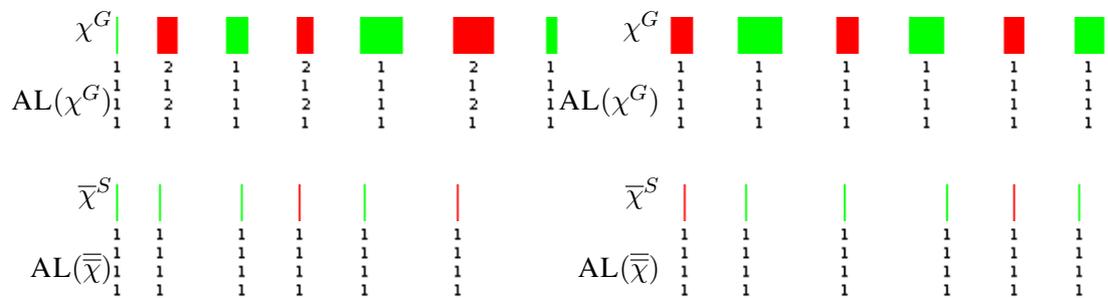
(25) p05\_arms\_slower

(26) p05\_arms\_worse



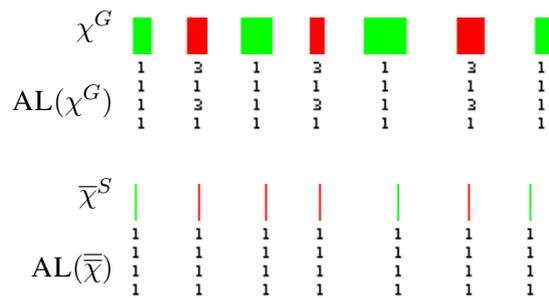
(27) p08\_arms\_normal02

(28) p08\_arms\_normal03



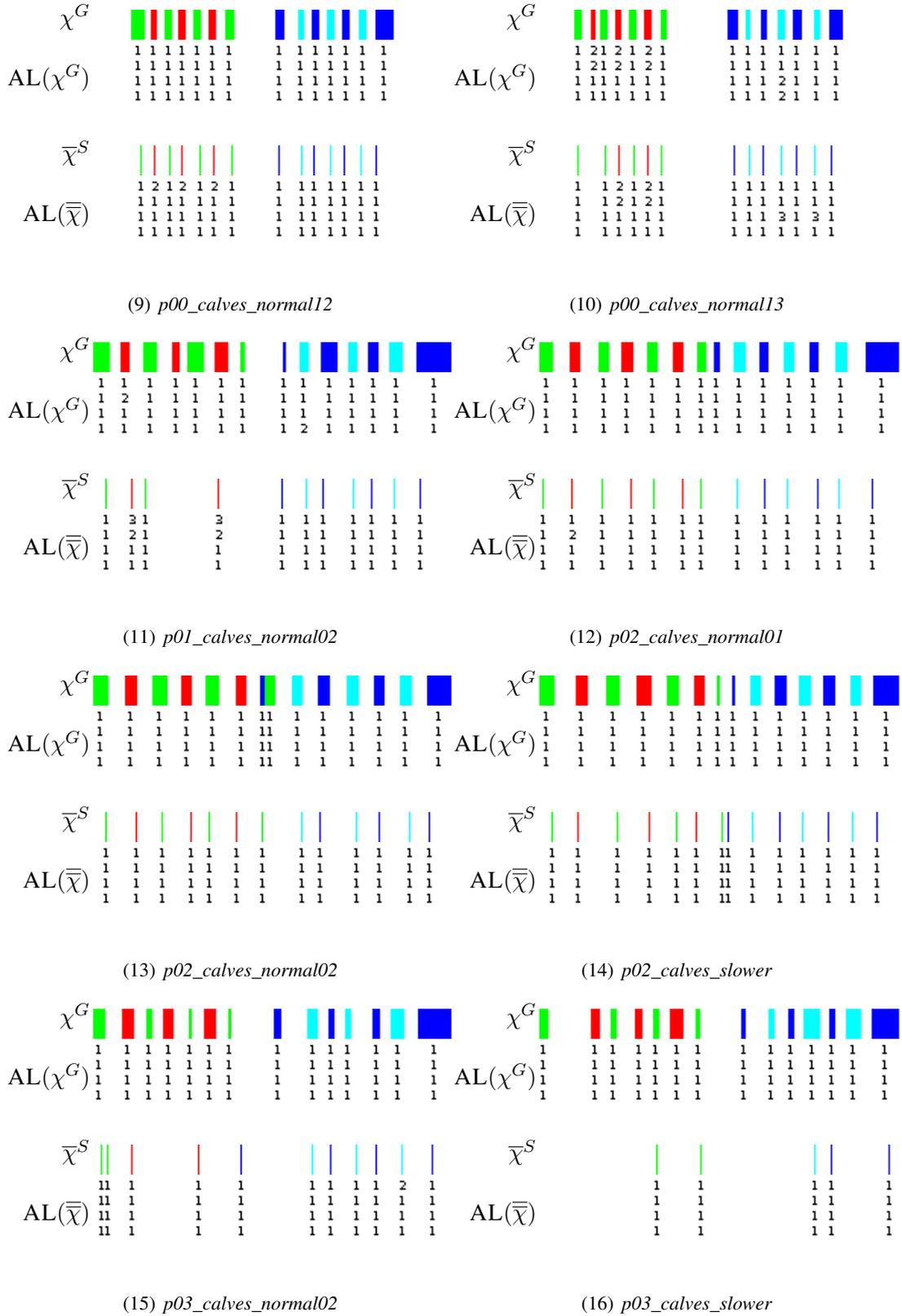
(29) p09\_arms\_normal01

(30) p09\_arms\_normal02



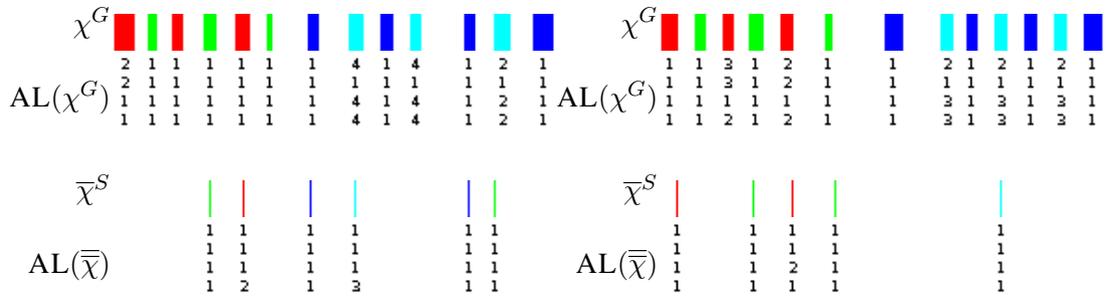
(31) p09\_arms\_worse

FIGURE B.3: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Calves* exercise (continued from previous page)



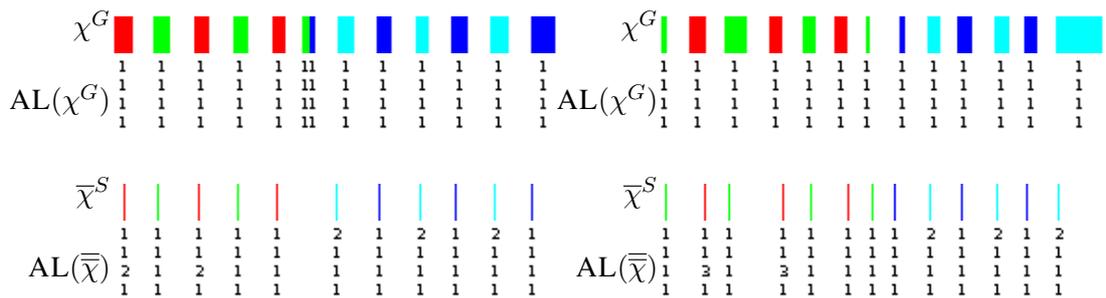
Continued on next page

FIGURE B.3: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Calves* exercise (continued from previous page)



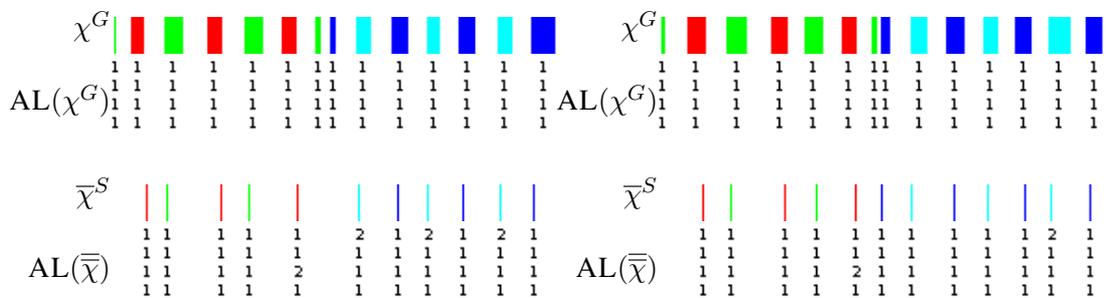
(17) *p05\_calves\_faster*

(18) *p05\_calves\_normal01*



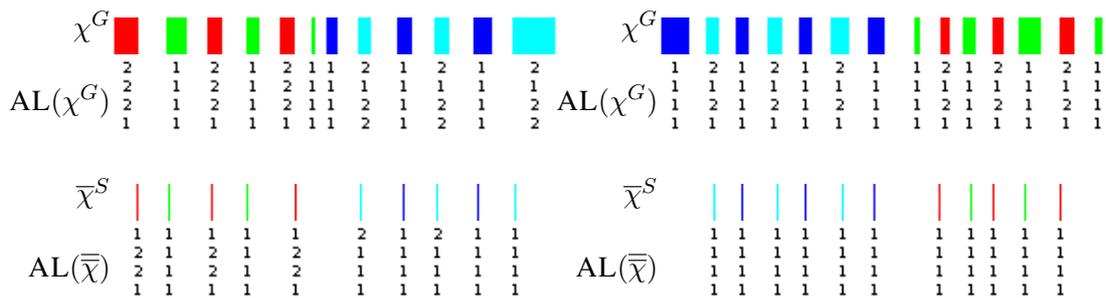
(19) *p06\_calves\_faster*

(20) *p06\_calves\_normal01*



(21) *p06\_calves\_normal02*

(22) *p06\_calves\_slower*

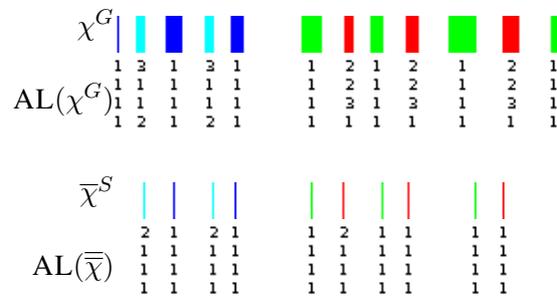


(23) *p06\_calves\_worse*

(24) *p09\_calves\_normal*

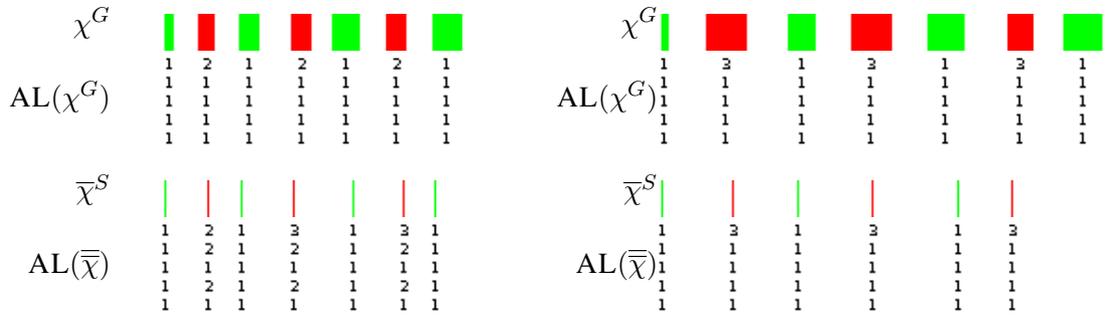
Continued on next page

FIGURE B.3: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Calves* exercise (continued from previous page)



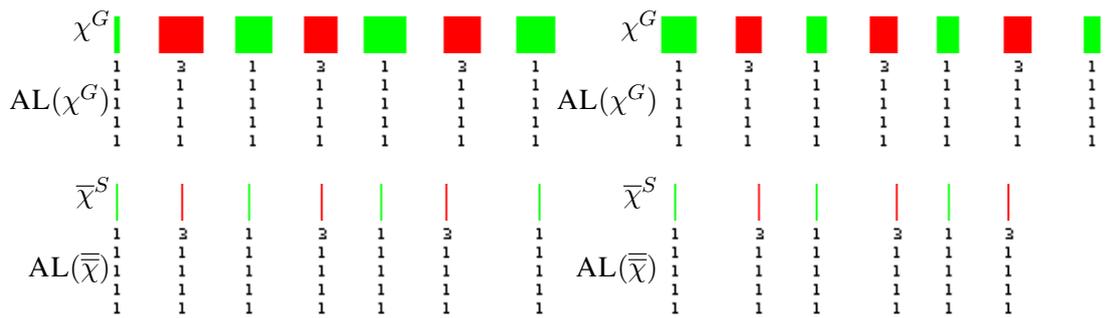
(25) *p09\_calves\_worse*

FIGURE B.4: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Inner thighs* exercise (continued from previous page)



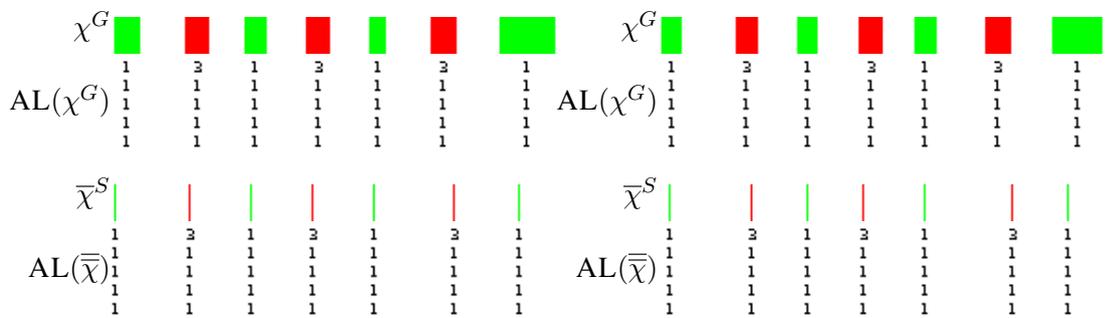
(9) *p00\_inner\_thighs\_normal11*

(10) *p01\_inner\_thighs\_normal01*



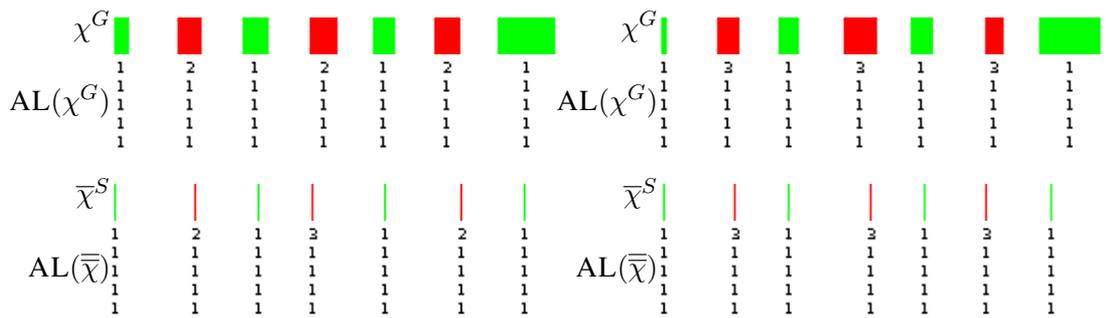
(11) *p01\_inner\_thighs\_normal02*

(12) *p02\_inner\_thighs\_normal01*



(13) *p02\_inner\_thighs\_normal02*

(14) *p02\_inner\_thighs\_slower*

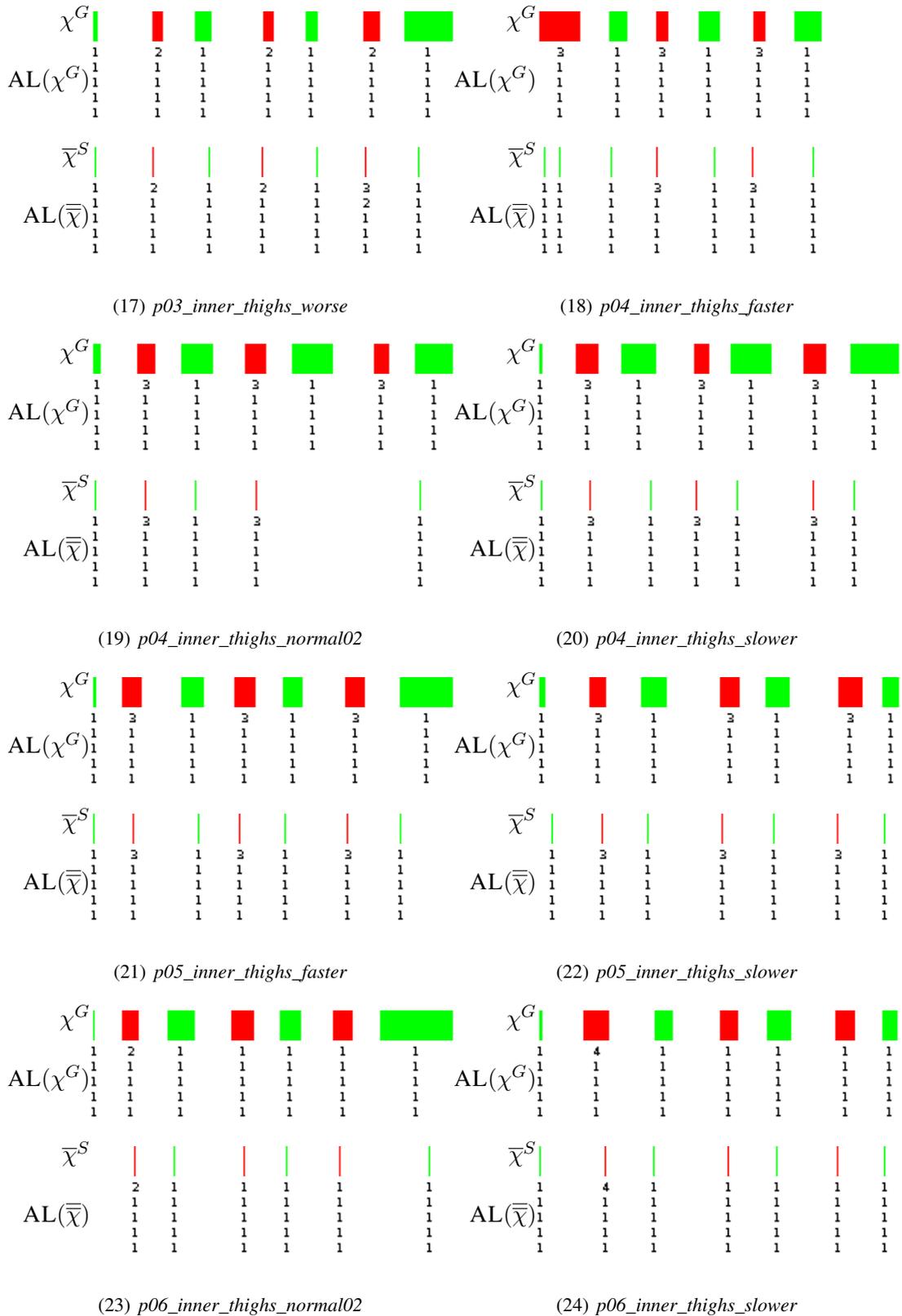


(15) *p02\_inner\_thighs\_worse*

(16) *p03\_inner\_thighs\_normal01*

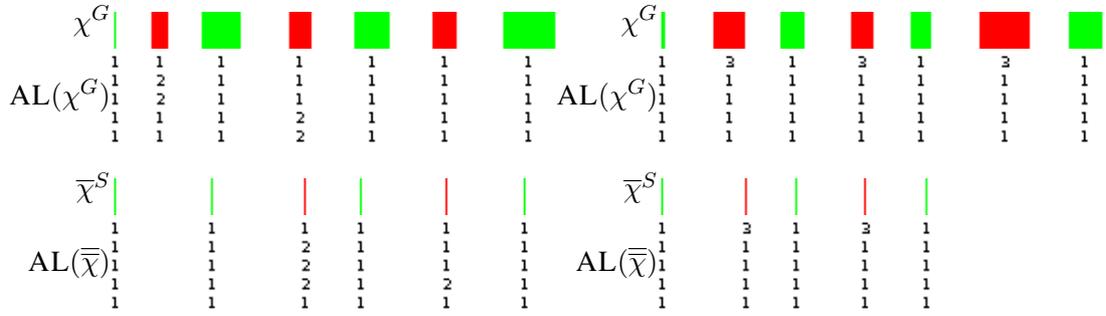
Continued on next page

FIGURE B.4: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Inner thighs* exercise (continued from previous page)

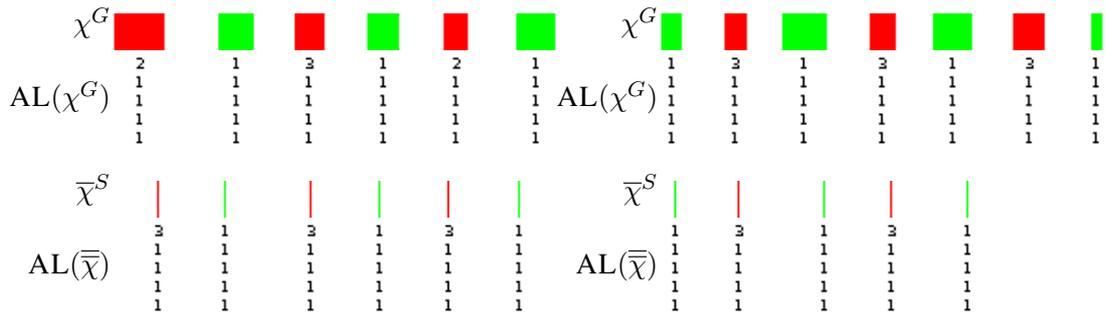


Continued on next page

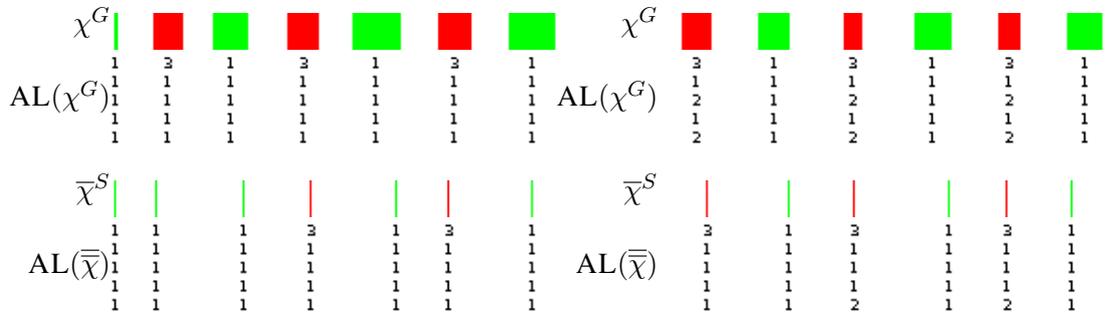
FIGURE B.4: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Inner thighs* exercise (continued from previous page)



(25) *p06\_inner\_thighs\_worse* (26) *p07\_inner\_thighs\_normal01*

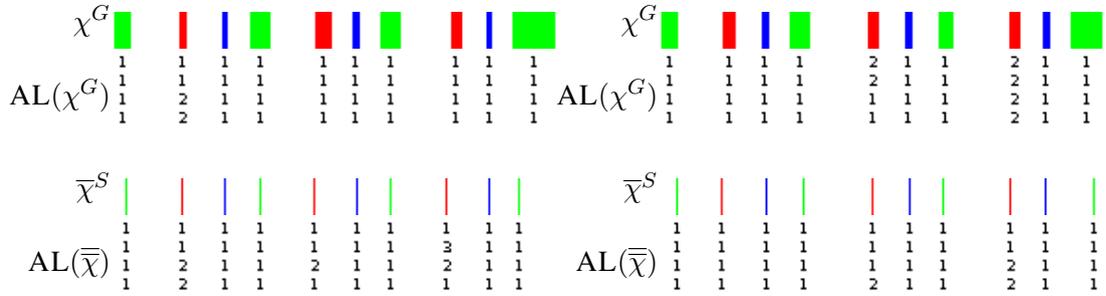


(27) *p08\_inner\_thighs\_normal02* (28) *p09\_inner\_thighs\_normal0*



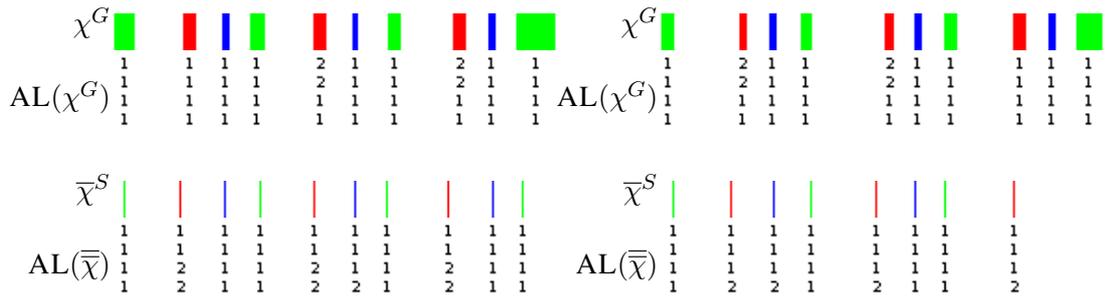
(29) *p09\_inner\_thighs\_normal01* (30) *p09\_inner\_thighs\_worse*

FIGURE B.5: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Shoulders* exercise (continued from previous page)



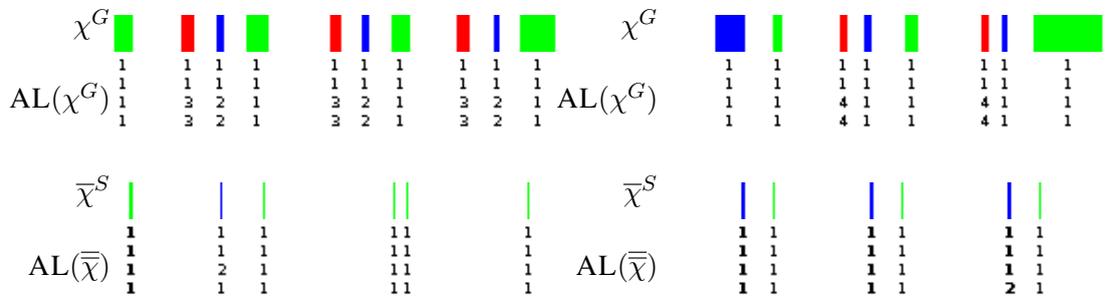
(9) *p01\_shoulders\_slower*

(10) *p02\_shoulders\_normal01*



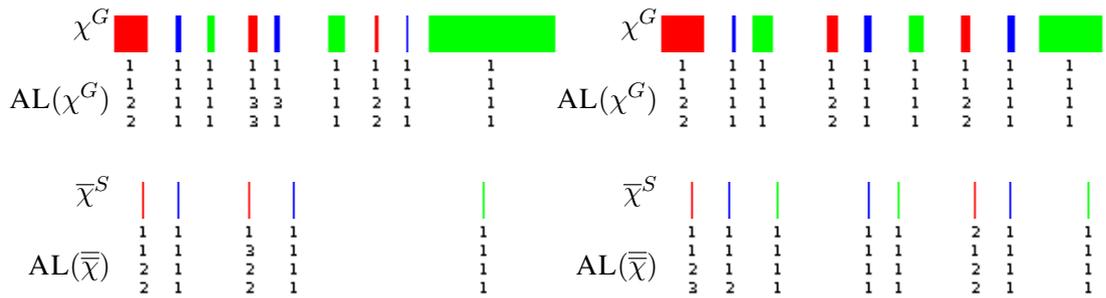
(11) *p02\_shoulders\_normal02*

(12) *p02\_shoulders\_slower*



(13) *p02\_shoulders\_worse*

(14) *p03\_shoulders\_worse*

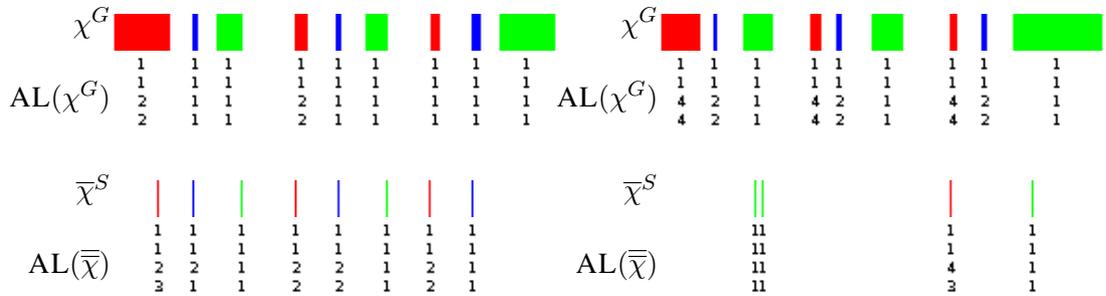


(15) *p04\_shoulders\_faster*

(16) *p04\_shoulders\_normal02*

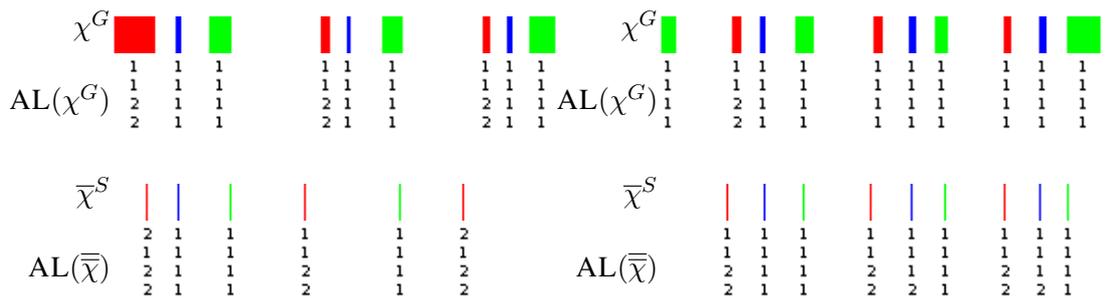
Continued on next page

FIGURE B.5: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Shoulders* exercise (continued from previous page)



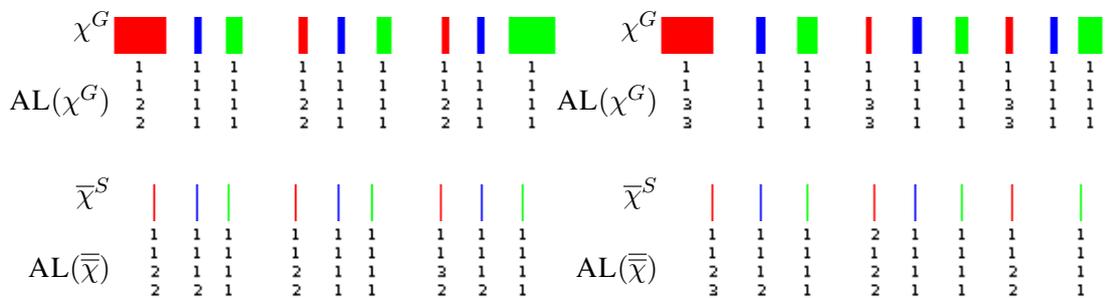
(17) *p04\_shoulders\_slower*

(18) *p04\_shoulders\_worse*



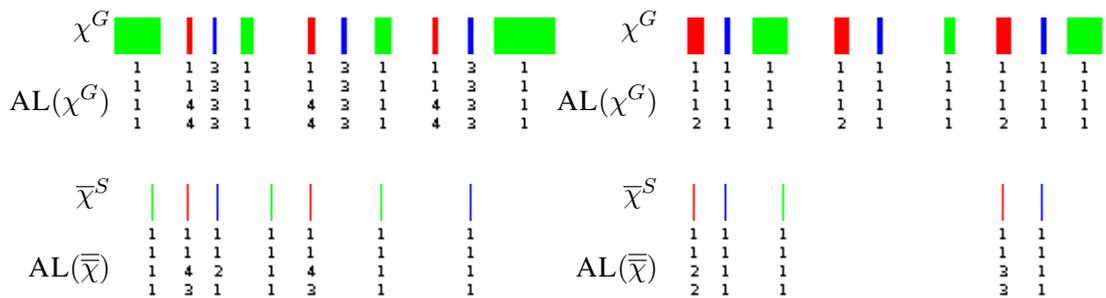
(19) *p05\_shoulders\_slower*

(20) *p06\_shoulders\_normal01*



(21) *p06\_shoulders\_normal02*

(22) *p06\_shoulders\_slower*



(23) *p06\_shoulders\_worse*

(24) *p09\_shoulders\_normal01*

Continued on next page

FIGURE B.5: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Shoulders* exercise (continued from previous page)

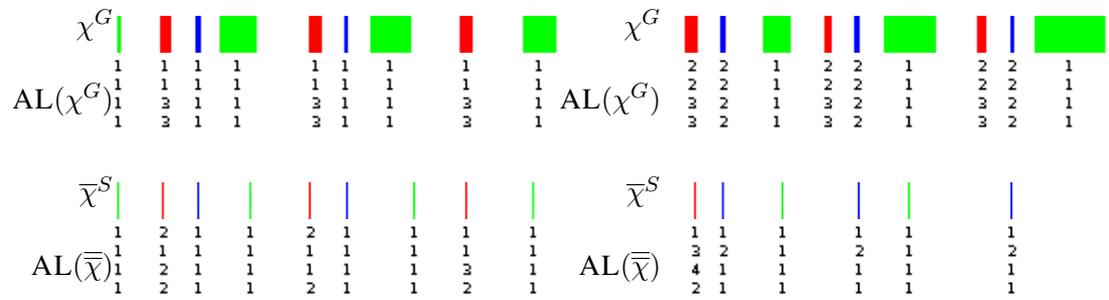
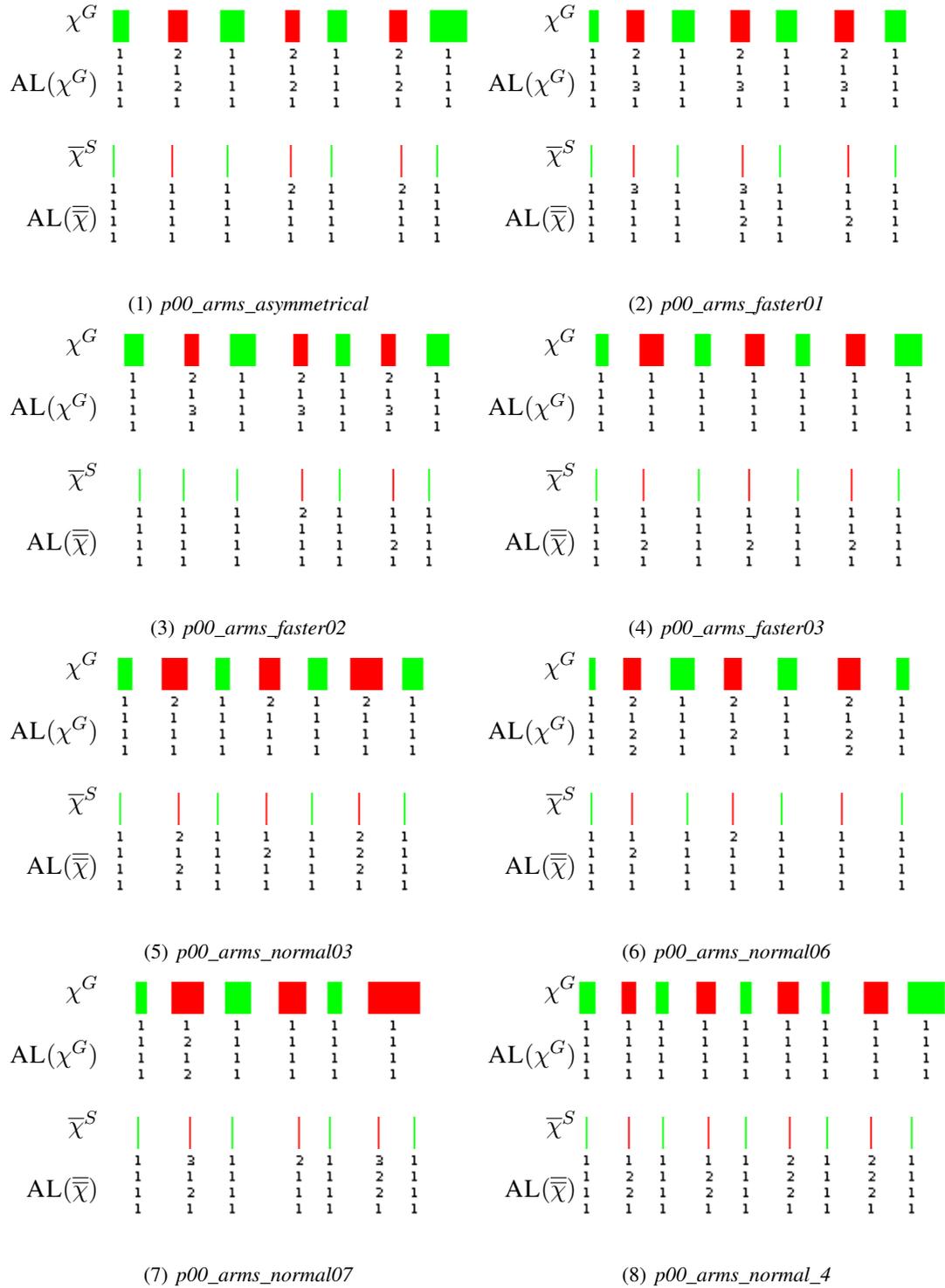
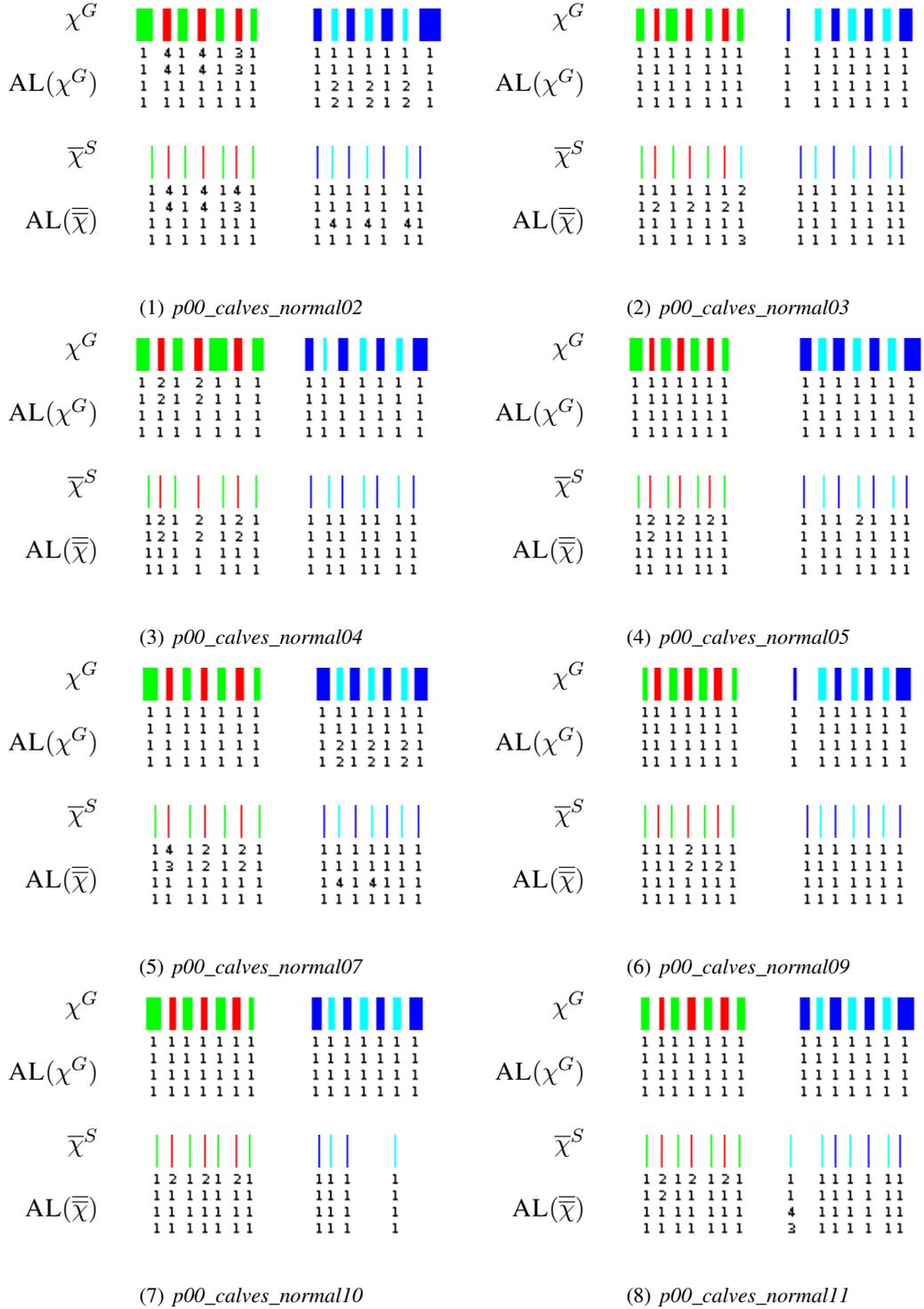


FIGURE B.2: Representation of results for periodic landmark sequence analysis and landmark assessment in the Arms exercise



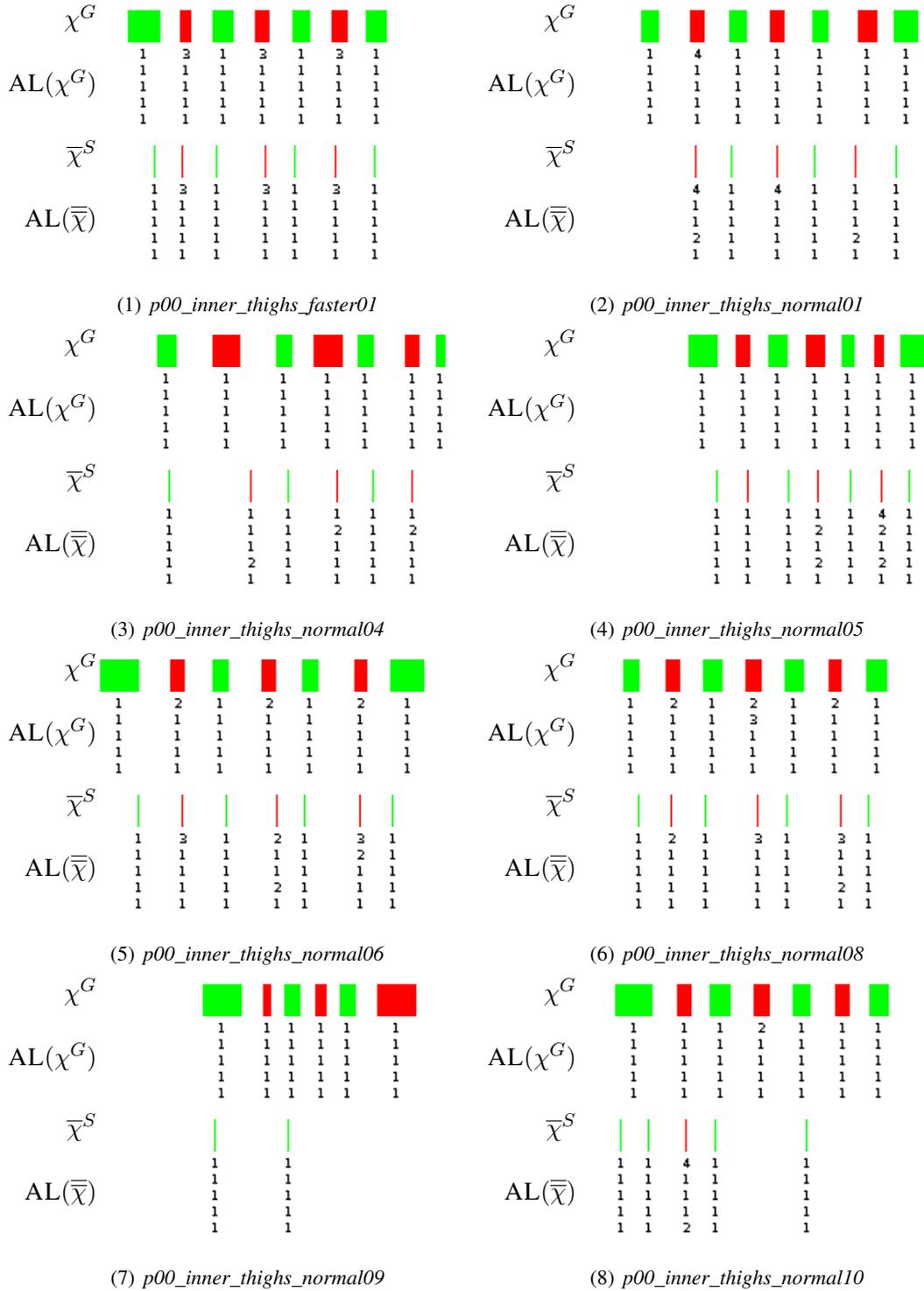
Continued on next page

FIGURE B.3: Representation of results for periodic landmark sequence analysis and landmark assessment in the Calves exercise



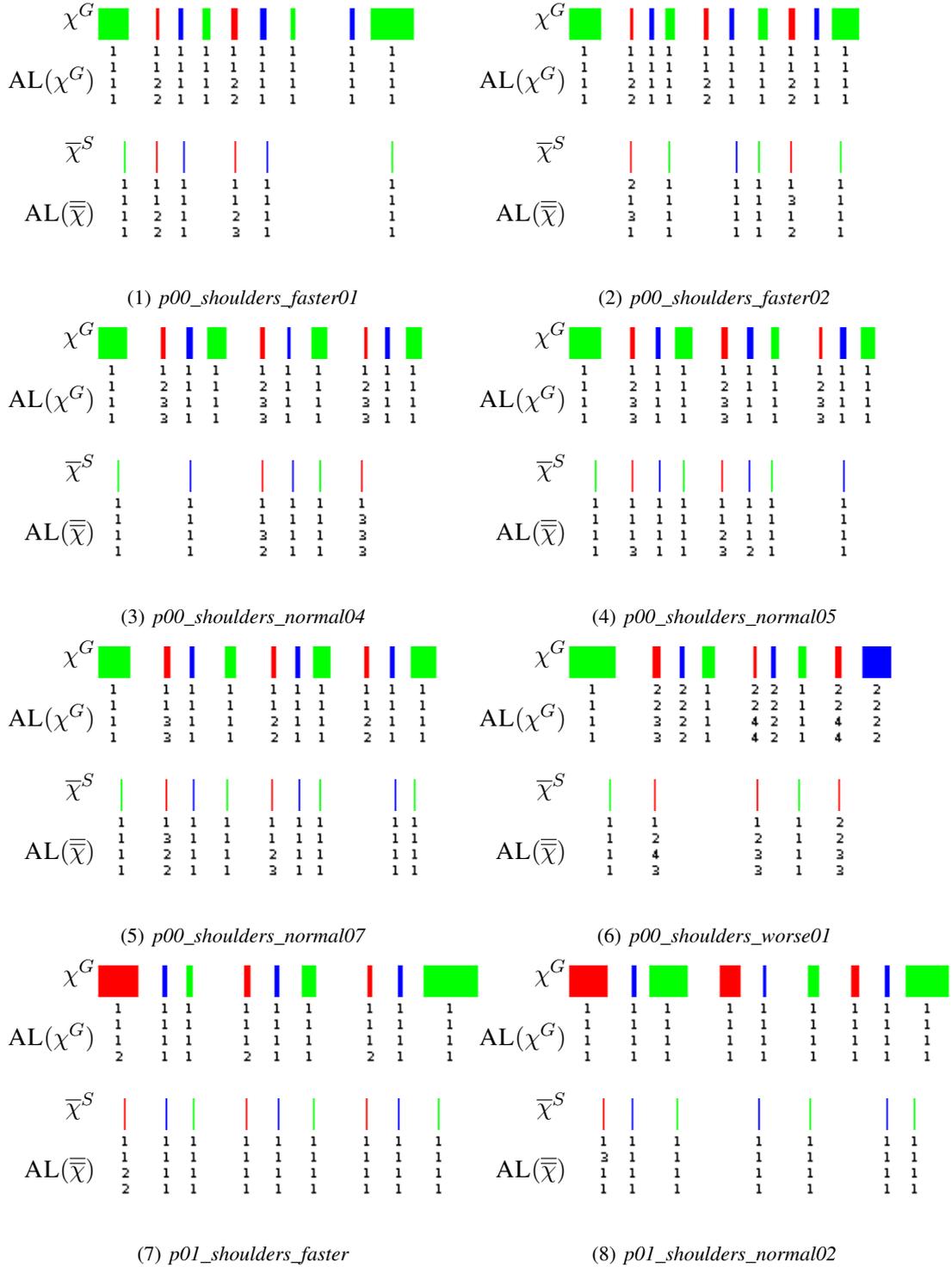
Continued on next page

FIGURE B.4: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Inner thighs* exercise



Continued on next page

FIGURE B.5: Representation of results for periodic landmark sequence analysis and landmark assessment in the *Shoulders* exercise



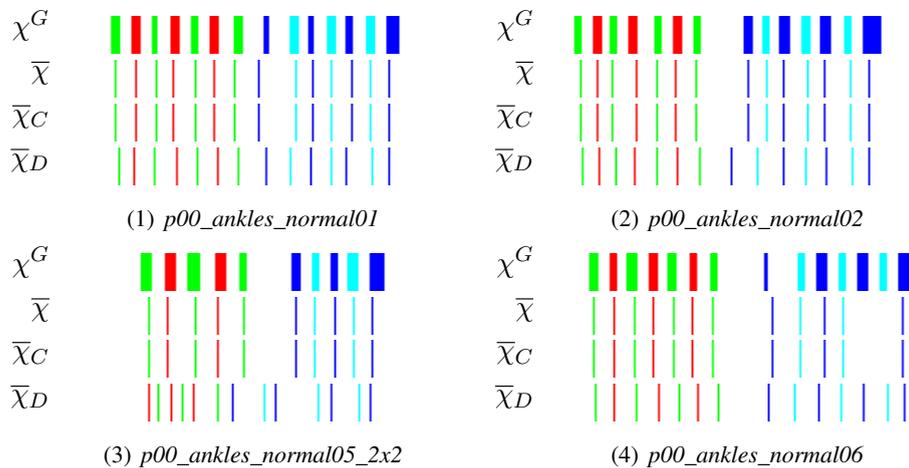
Continued on next page

## Appendix C

# Comparison of results for landmark identification and landmark refining with Dynamic Time Warping (DTW)

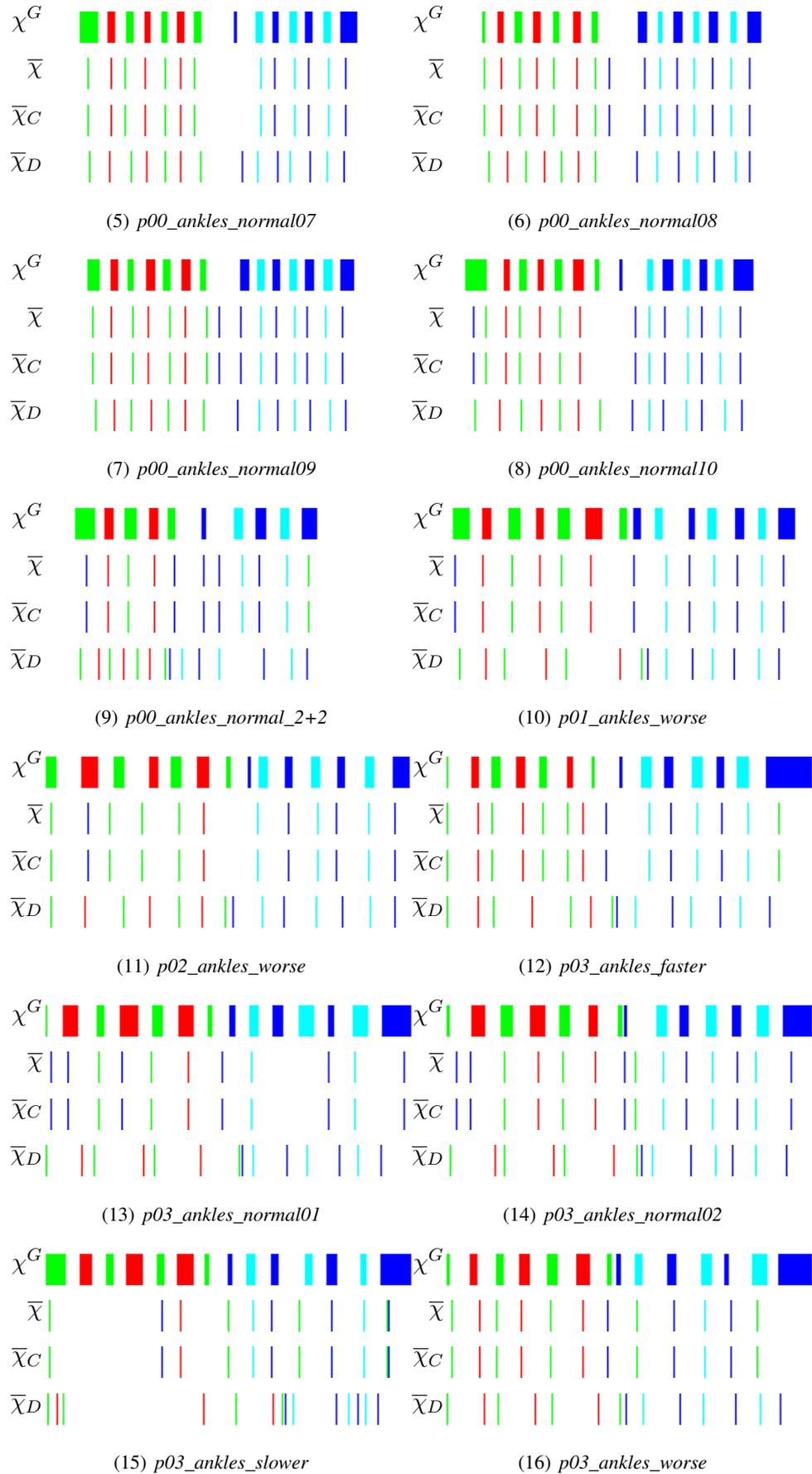
Comparison of results between landmark identification+landmark refining and DTW. Ground truth landmarks ( $\chi^G$ ), refined landmarks using a machine learning classifier+cluster analysis ( $\bar{\chi}_C$ ) and DTW ( $\bar{\chi}_D$ ) represented with vertical markers of different colours, each of which representing a ground-truth, refined or identified landmark type, respectively.

FIGURE C.1: Comparison of results between landmark identification+landmark refining and DTW in the *Ankles* exercise



Continued on next page

FIGURE C.1: Comparison of results between landmark identification+landmark refining and DTW in the *Ankles* exercise (continued from previous page)



Continued on next page

FIGURE C.1: Comparison of results between landmark identification+landmark refining and DTW in the *Ankles* exercise (continued from previous page)

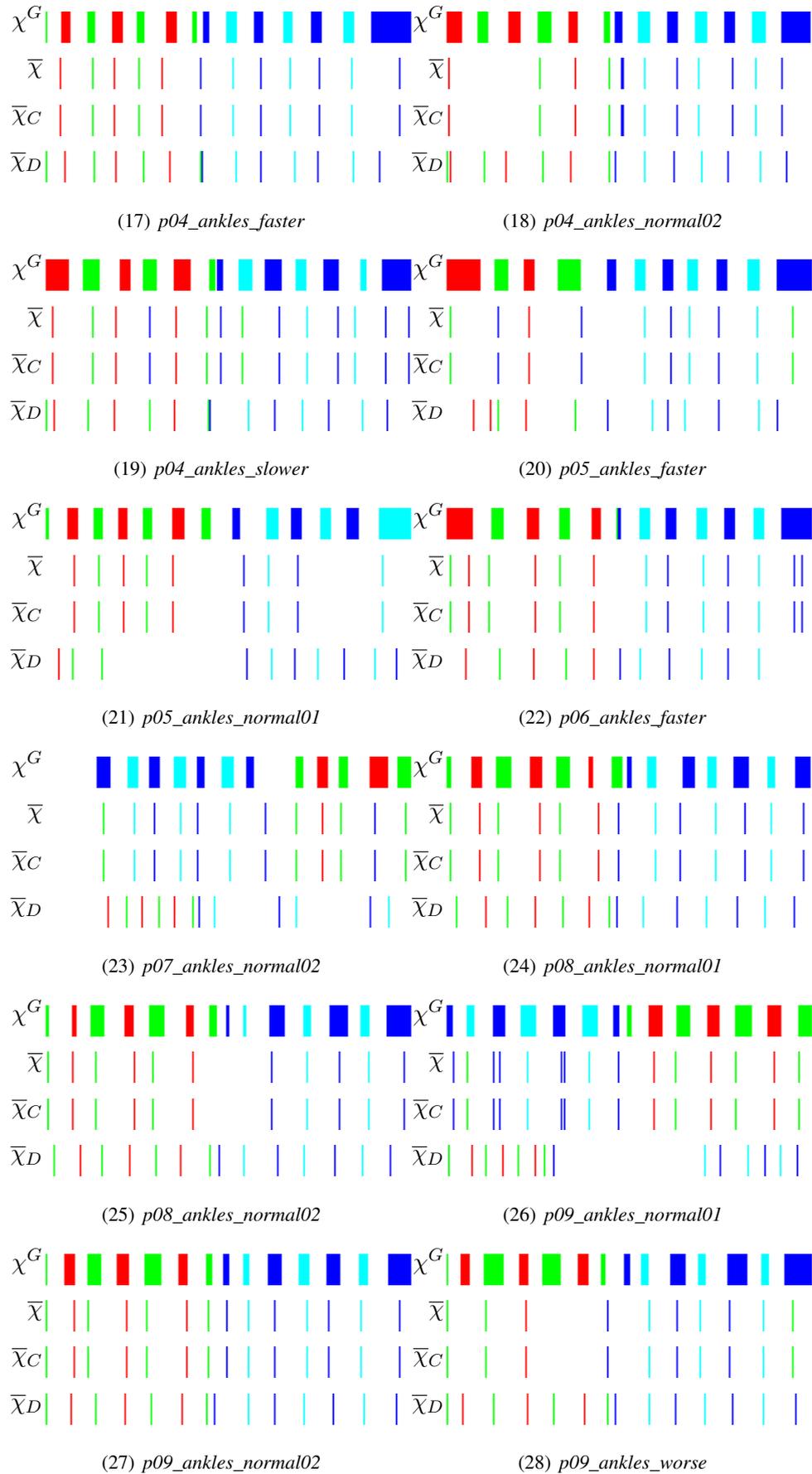
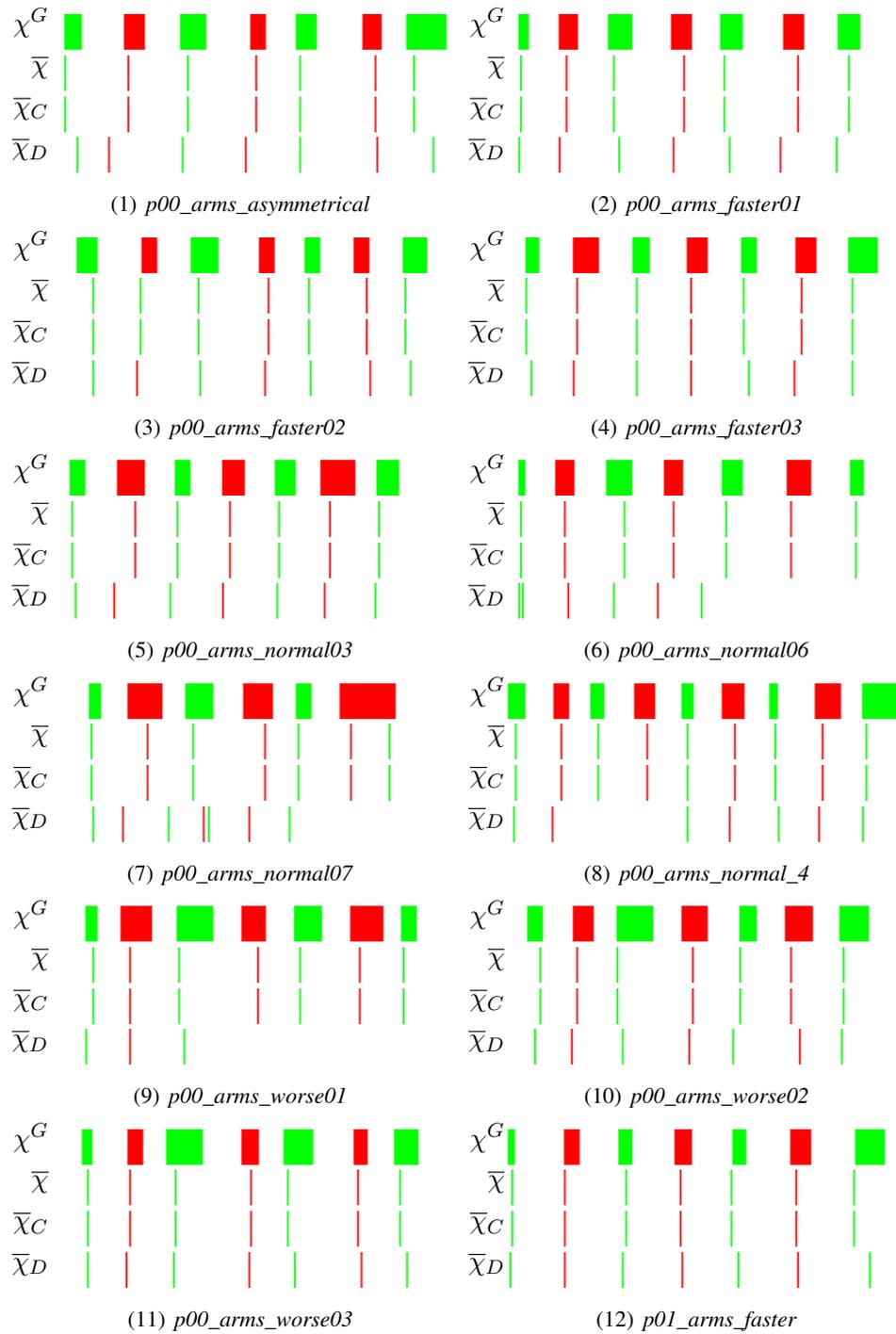
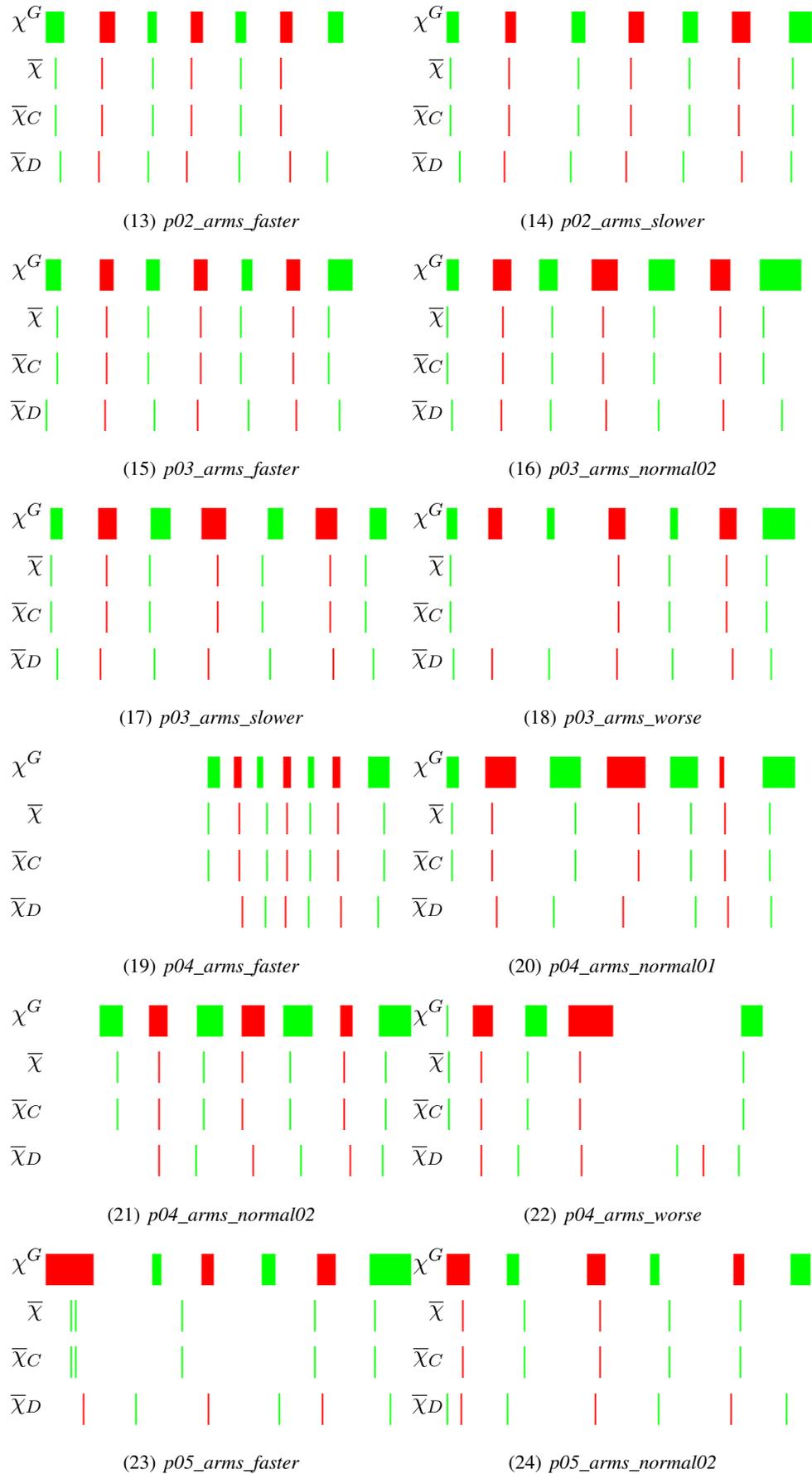


FIGURE C.2: Comparison of results between landmark identification+landmark refining and DTW in the Arms exercise



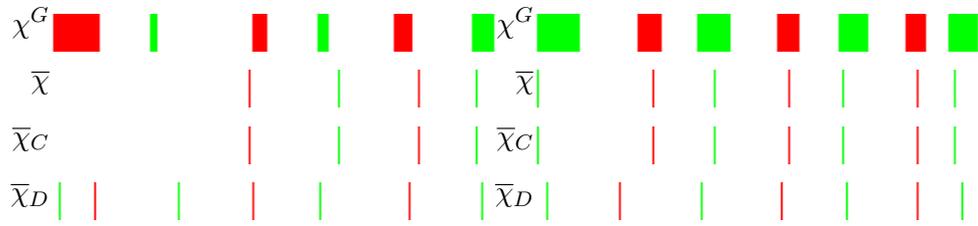
Continued on next page

FIGURE C.2: Comparison of results between landmark identification+landmark refining and DTW in the Arms exercise (continued from previous page)



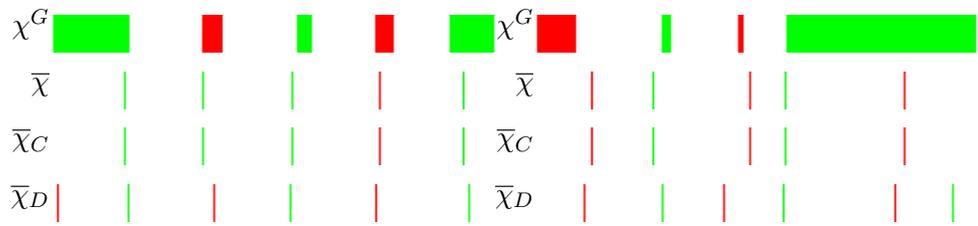
Continued on next page

FIGURE C.2: Comparison of results between landmark identification+landmark refining and DTW in the Arms exercise (continued from previous page)



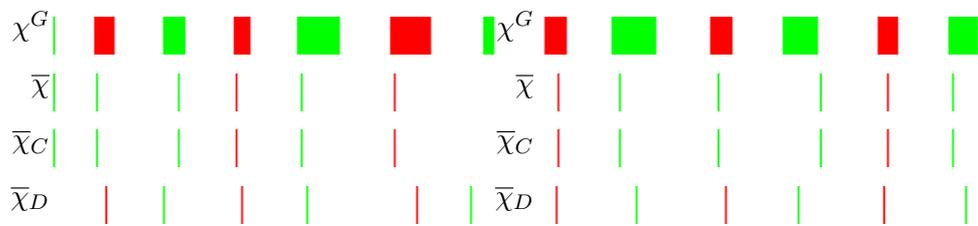
(25) *p05\_arms\_slower*

(26) *p05\_arms\_worse*



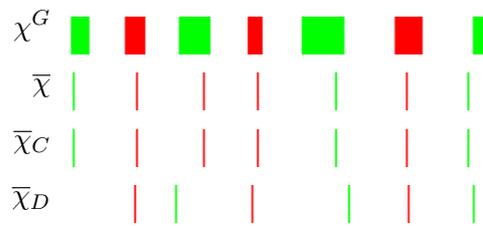
(27) *p08\_arms\_normal02*

(28) *p08\_arms\_normal03*



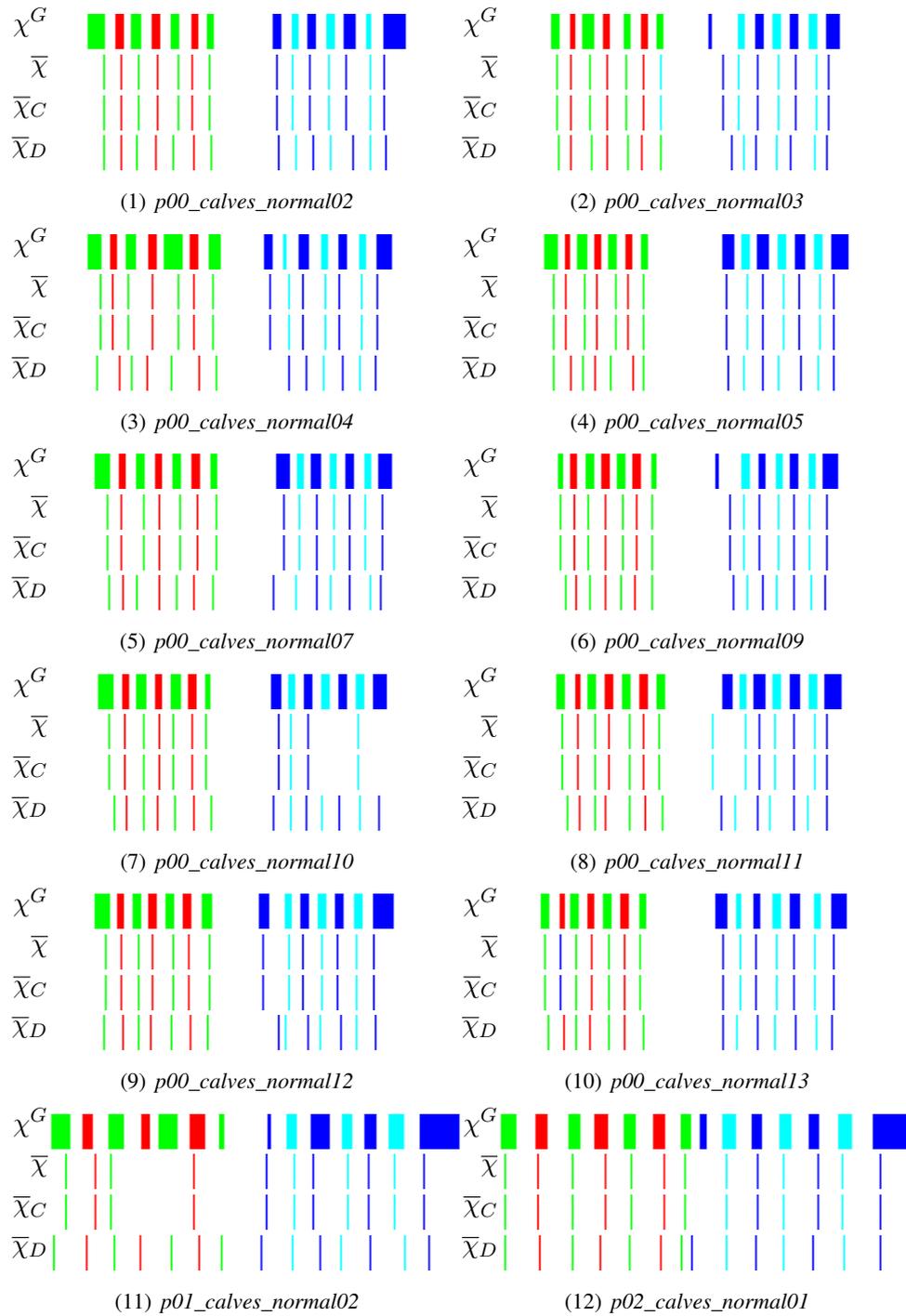
(29) *p09\_arms\_normal01*

(30) *p09\_arms\_normal02*



(31) *p09\_arms\_worse*

FIGURE C.3: Comparison of results between landmark identification+landmark refining and DTW in the Calves exercise



Continued on next page

FIGURE C.3: Comparison of results between landmark identification+landmark refining and DTW in the Calves exercise (continued from previous page)

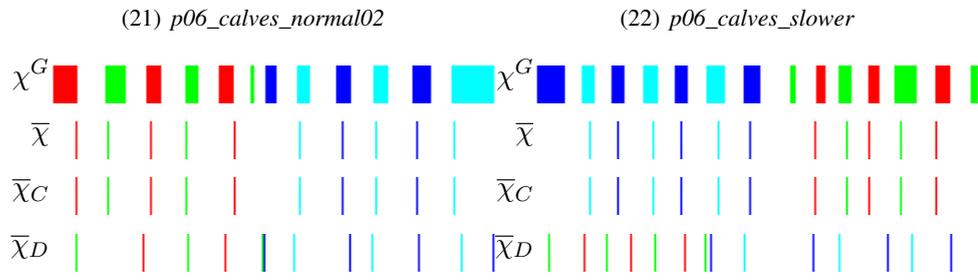
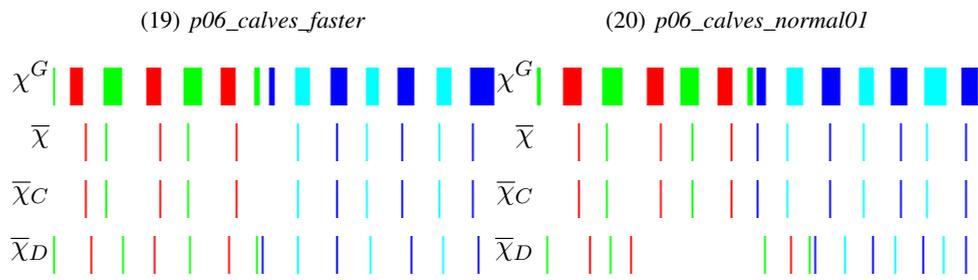
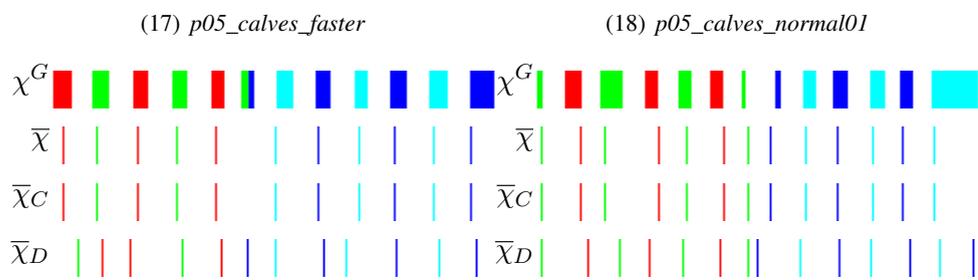
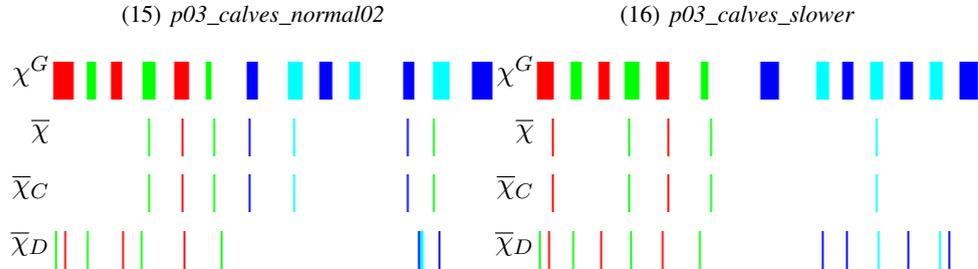
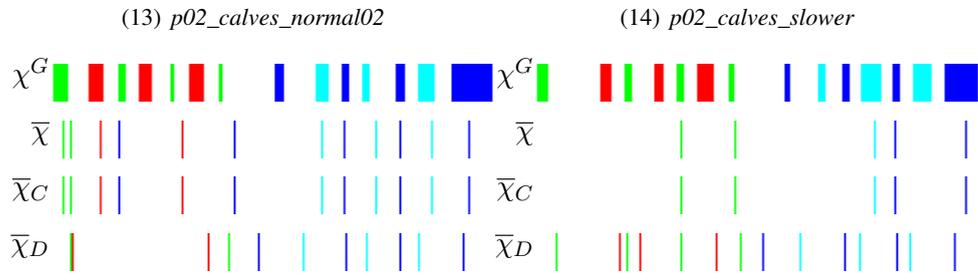
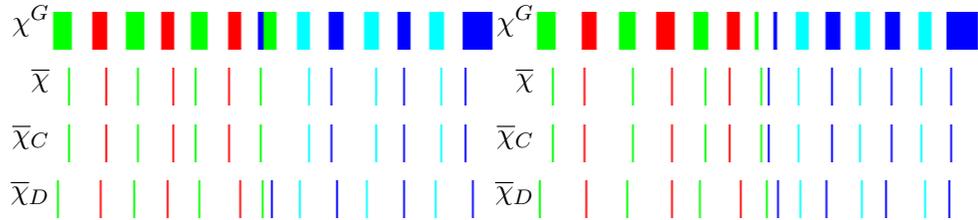
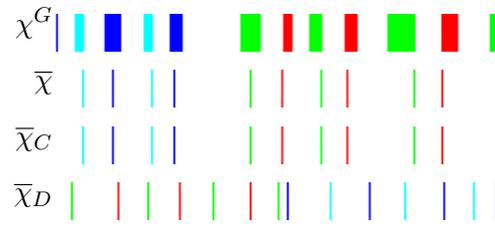
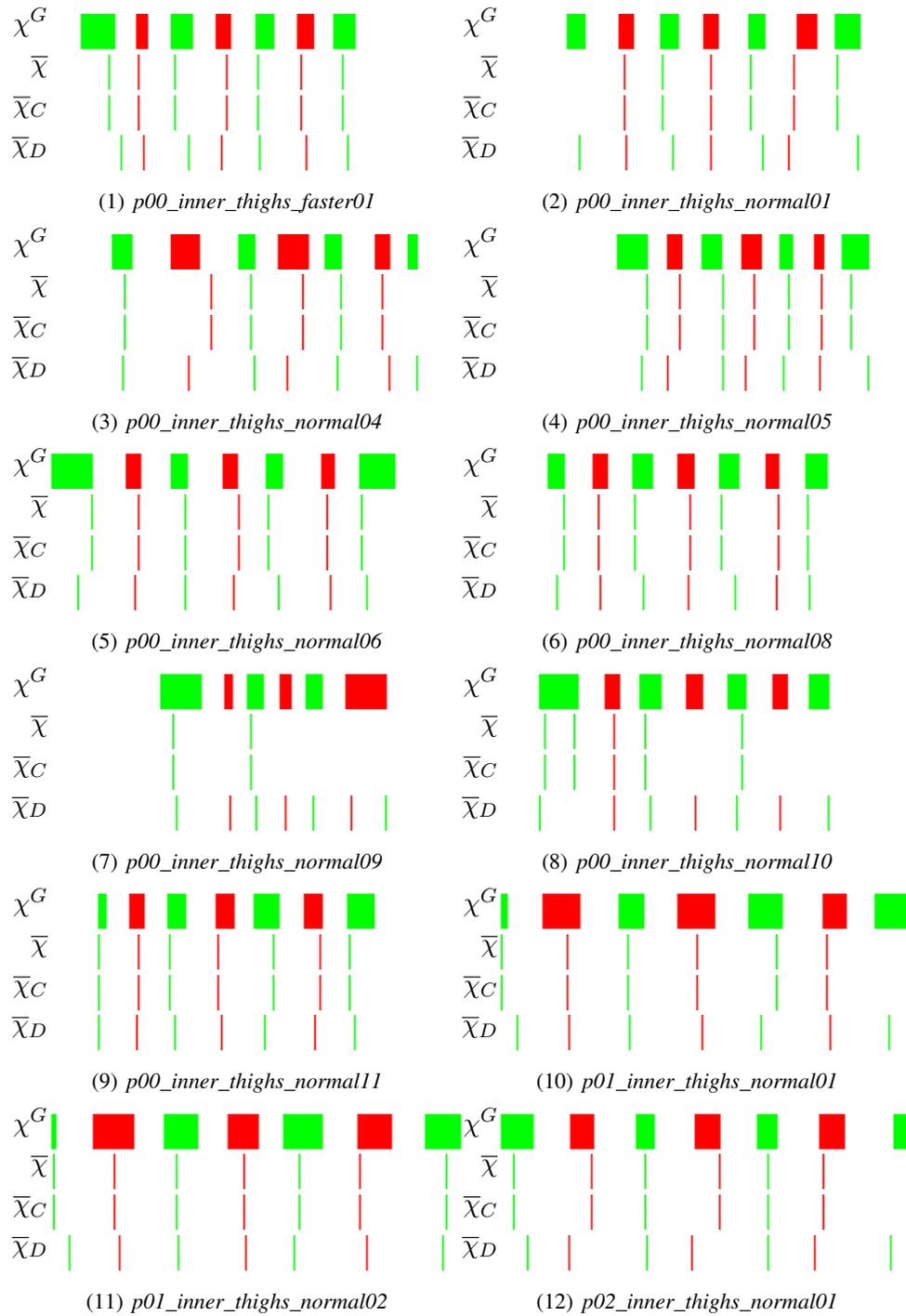


FIGURE C.3: Comparison of results between landmark identification+landmark refining and DTW in the *Calves* exercise (continued from previous page)



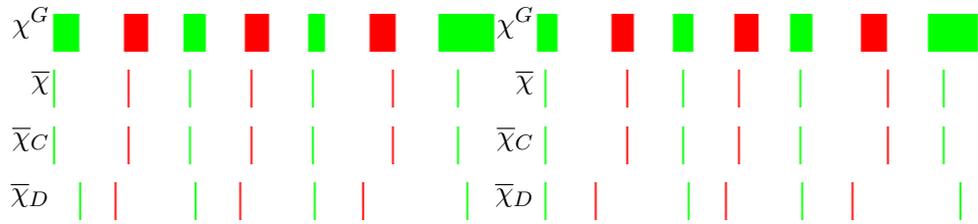
(25) *p09\_calves\_worse*

FIGURE C.4: Comparison of results between landmark identification+landmark refining and DTW in the *Inner thighs* exercise



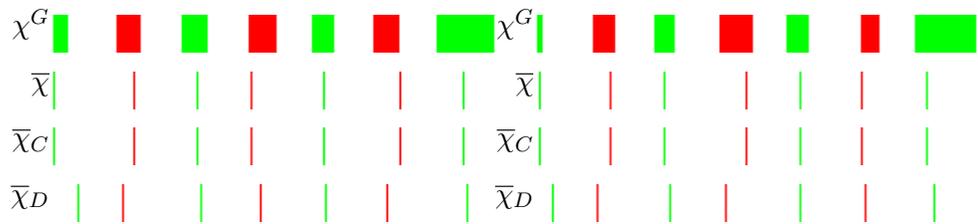
Continued on next page

FIGURE C.4: Comparison of results between landmark identification+landmark refining and DTW in the *Inner thighs* exercise (continued from previous page)



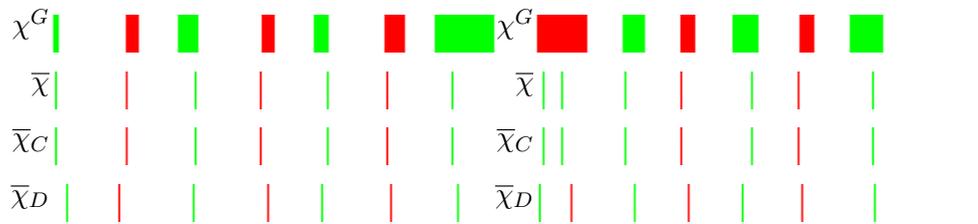
(13) *p02\_inner\_thighs\_normal02*

(14) *p02\_inner\_thighs\_slower*



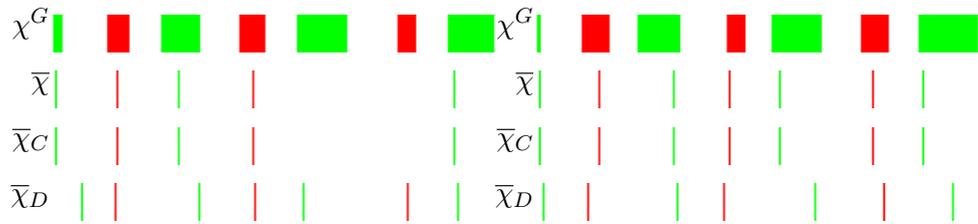
(15) *p02\_inner\_thighs\_worse*

(16) *p03\_inner\_thighs\_normal01*



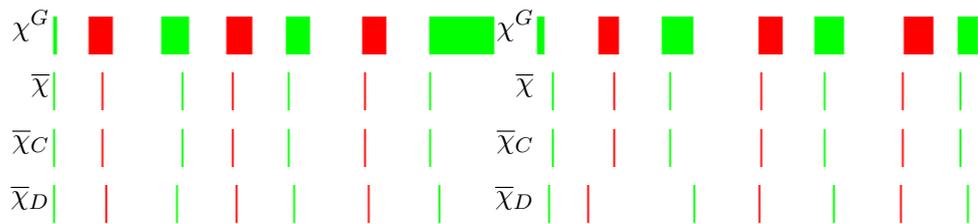
(17) *p03\_inner\_thighs\_worse*

(18) *p04\_inner\_thighs\_faster*



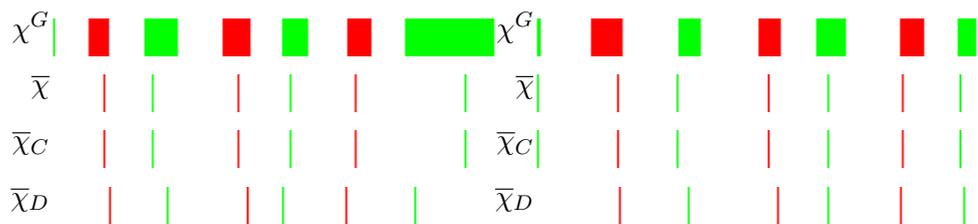
(19) *p04\_inner\_thighs\_normal02*

(20) *p04\_inner\_thighs\_slower*



(21) *p05\_inner\_thighs\_faster*

(22) *p05\_inner\_thighs\_slower*



(23) *p06\_inner\_thighs\_normal02*

(24) *p06\_inner\_thighs\_slower*

Continued on next page

FIGURE C.4: Comparison of results between landmark identification+landmark refining and DTW in the *Inner thighs* exercise (continued from previous page)

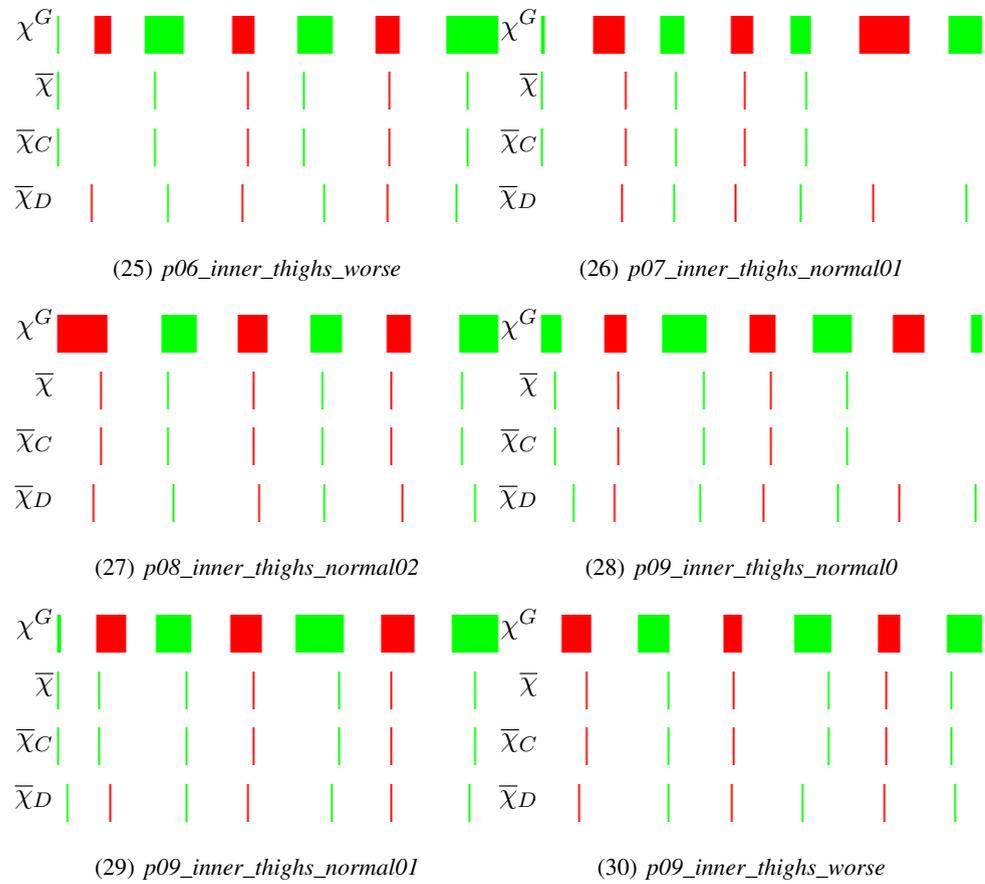
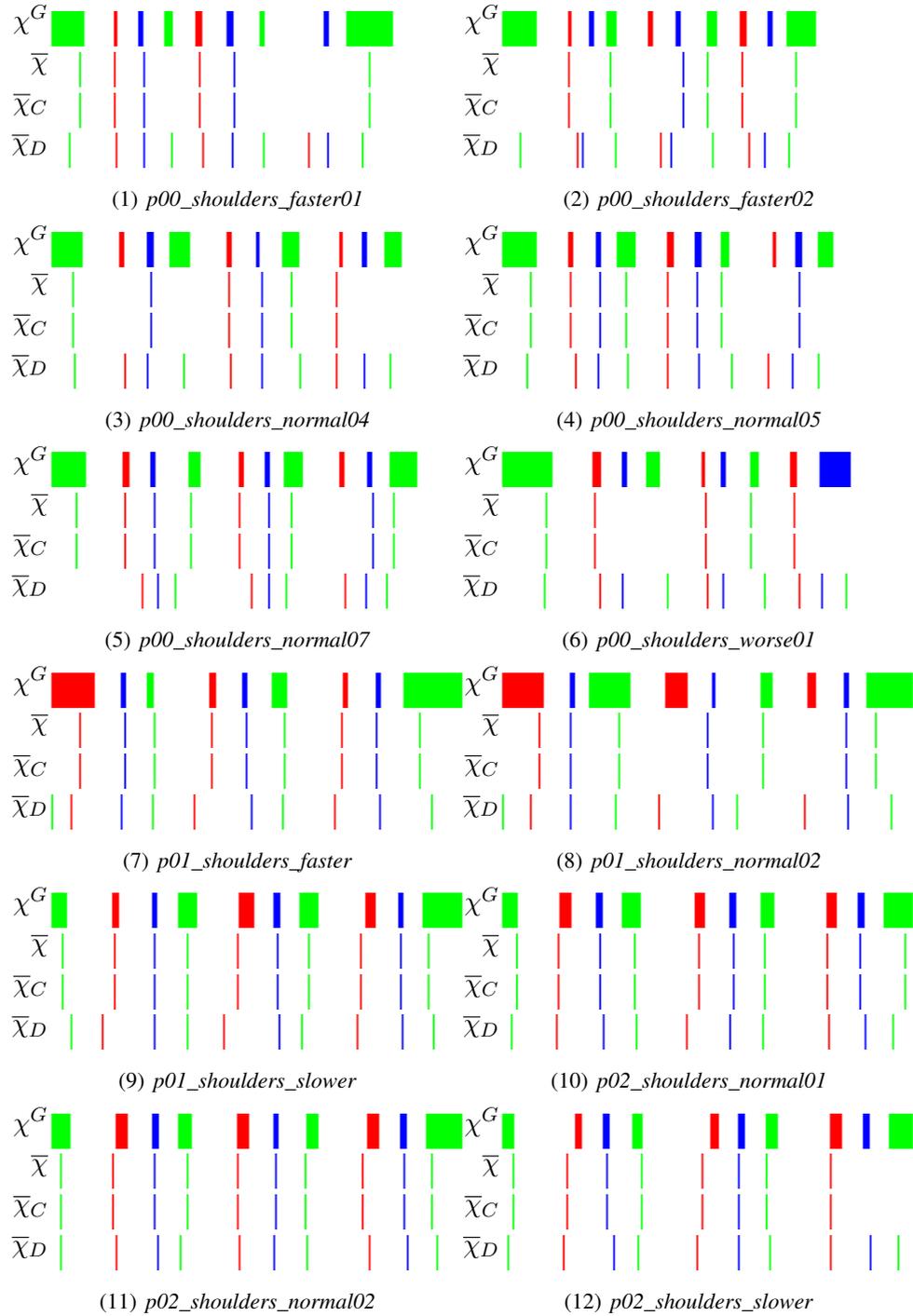
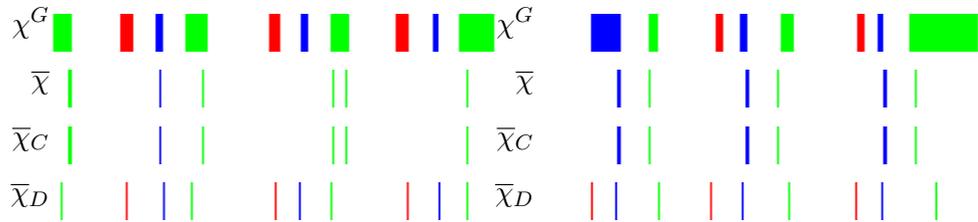


FIGURE C.5: Comparison of results between landmark identification+landmark refining and DTW in the *Shoulders* exercise



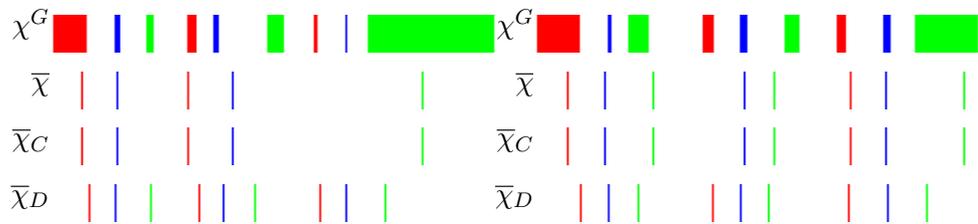
Continued on next page

FIGURE C.5: Comparison of results between landmark identification+landmark refining and DTW in the *Shoulders* exercise (continued from previous page)



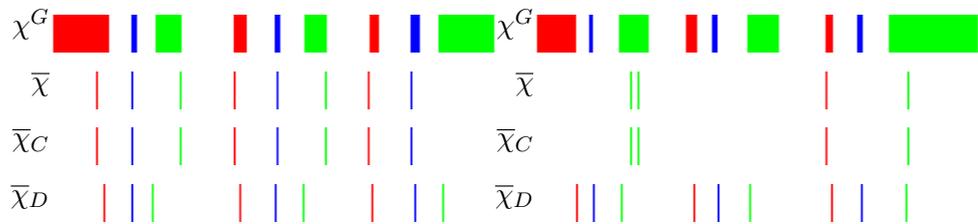
(13) *p02\_shoulders\_worse*

(14) *p03\_shoulders\_worse*



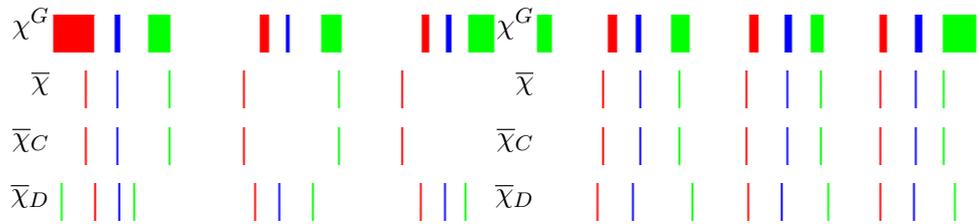
(15) *p04\_shoulders\_faster*

(16) *p04\_shoulders\_normal02*



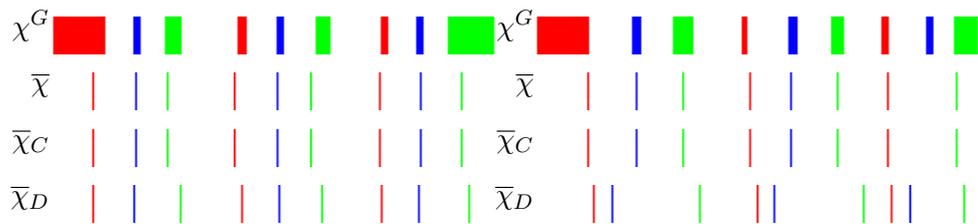
(17) *p04\_shoulders\_slower*

(18) *p04\_shoulders\_worse*



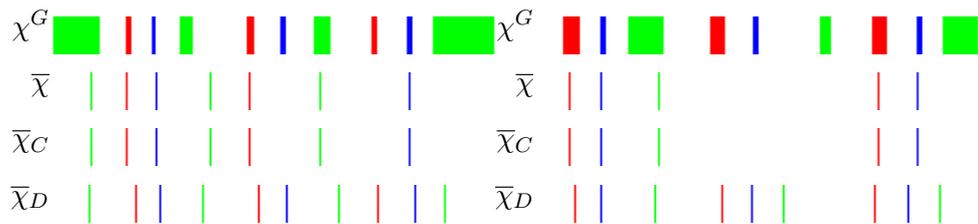
(19) *p05\_shoulders\_slower*

(20) *p06\_shoulders\_normal01*



(21) *p06\_shoulders\_normal02*

(22) *p06\_shoulders\_slower*

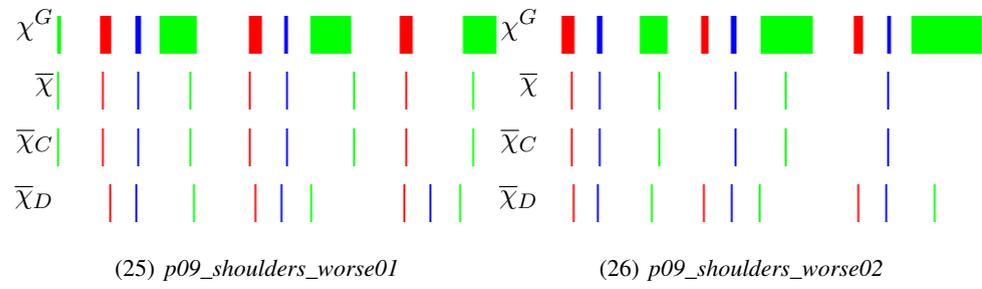


(23) *p06\_shoulders\_worse*

(24) *p09\_shoulders\_normal01*

Continued on next page

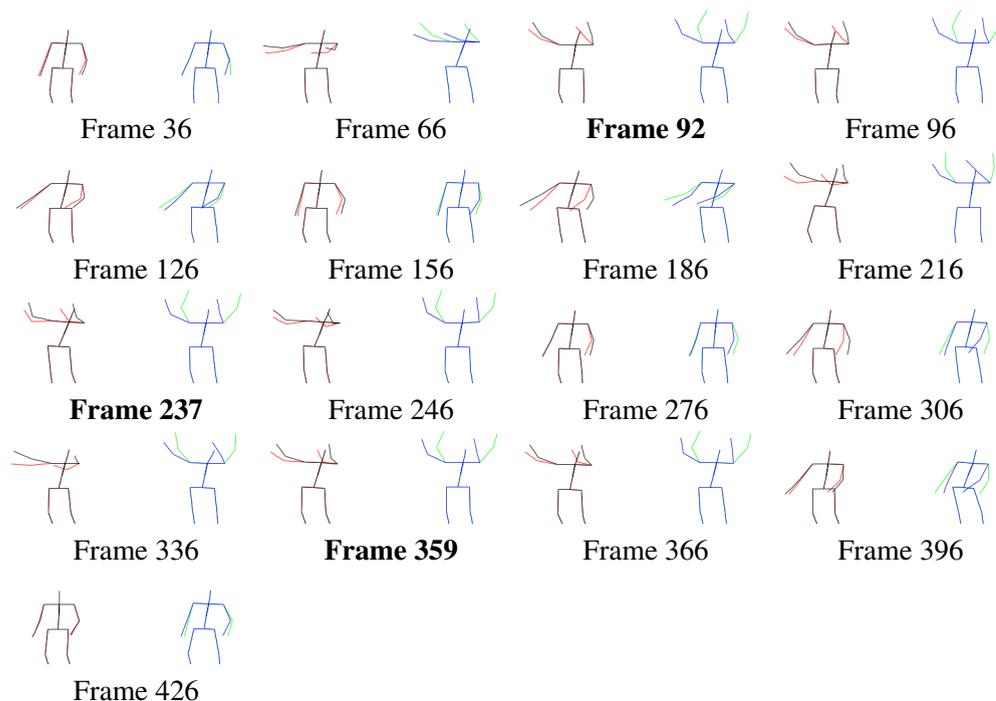
FIGURE C.5: Comparison of results between landmark identification+landmark refining and DTW in the *Shoulders* exercise (continued from previous page)



## **Appendix D**

# **Generated landmark assessment, motion synchronisation and motion adjustment**

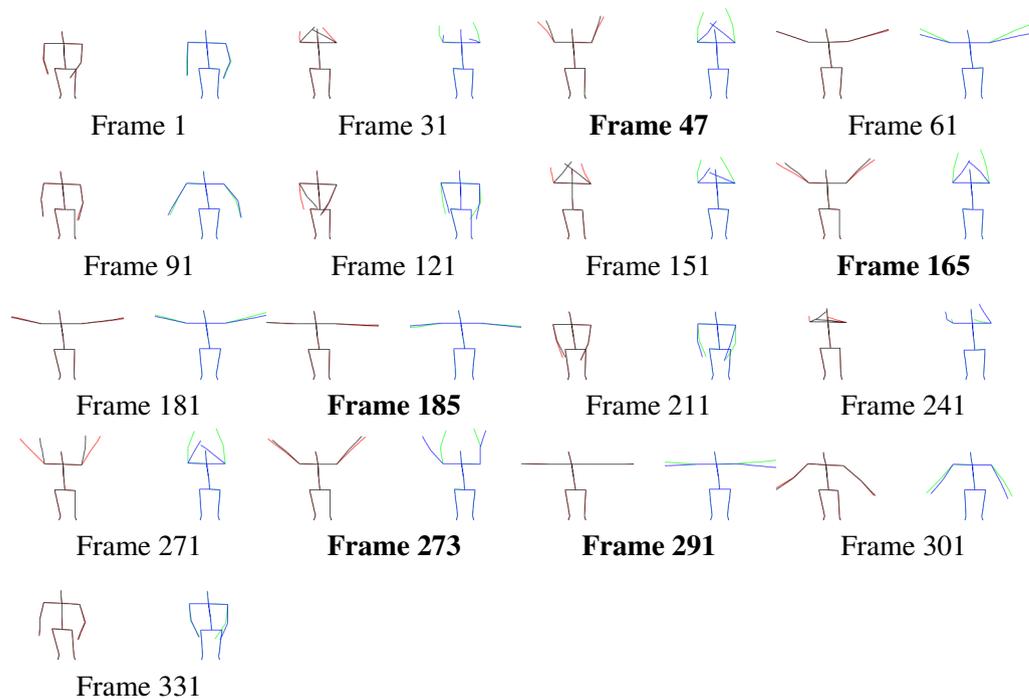
FIGURE D.1: Generated assessment and adjustment for *p00\_arms\_worse03*. Sequence of the reconstructed skeletons of the user's performance (in red) and its adjustment to the tutor's (in black) with  $\gamma = 0.5$ . Additionally, the synchronised animation of the tutor's demonstration (in green) and its adjustment to the user (in blue) with  $\gamma = 0.5$ . The animation is to be followed from left to right and top-bottom



Generated feedback:

- Frame 92: “*On this occasion, you are not rising your left arm. Furthermore, you are not rising your right arm. Look at the instructor and try to reach up with your arms on top of your head.*”
- Frame 237: “*Now, once more, you are not rising your left arm. Once more, you are not rising your right arm.*”
- Frame 359: “*This time, again, you are not rising your left arm. Again, you are not rising your right arm.*”

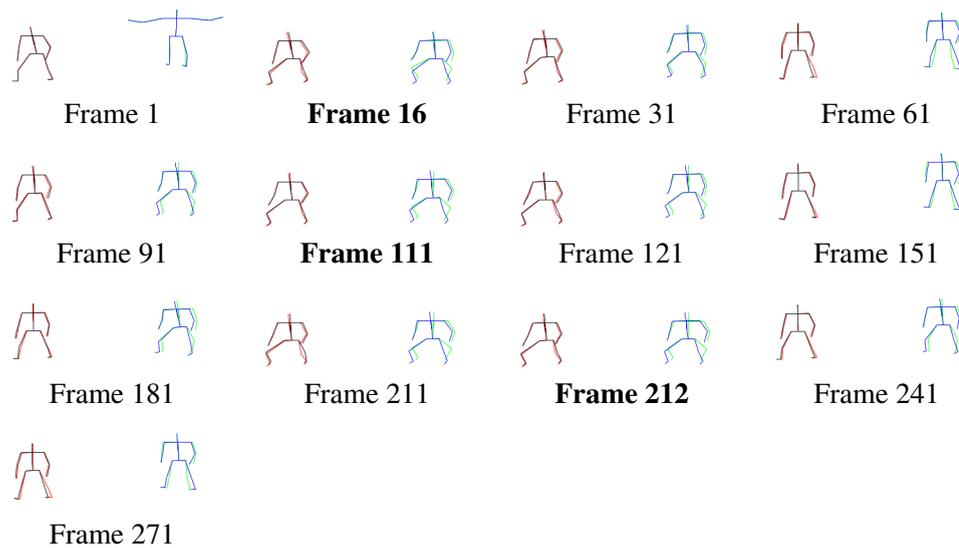
FIGURE D.2: Generated assessment and adjustment for *p01\_shoulders\_worse*. Sequence of the reconstructed skeletons of the user's performance (in red) and its adjustment to the tutor's (in black) with  $\gamma = 0.5$ . Additionally, the synchronised animation of the tutor's demonstration (in green) and its adjustment to the user (in blue) with  $\gamma = 0.5$ . The animation is to be followed from left to right and top-bottom



Generated feedback:

- Frame 47: *“On this occasion, your left arm should be raised much higher up. Additionally, your left arm should be stretched a little bit more. Look at the instructor and try to reach up with your arms on top of your head. And try to stretch your arms over your head, with forearm completely parallel to your arm.”*
- Frame 165: *“This time, again, your left arm should be raised much higher up. Once more, your left arm should be stretched a little bit more.”*
- Frame 185: *“On this occasion, your left arm needs to be stretched much more. Look at the instructor; your upper limbs should be stretched horizontally on a T pose.”*
- Frame 273: *“This time, again, your left arm should be raised much higher up. Again, your left arm should be stretched a little bit more.”*
- Frame 291: *“On this occasion, once more, your left arm needs to be stretched much more.”*

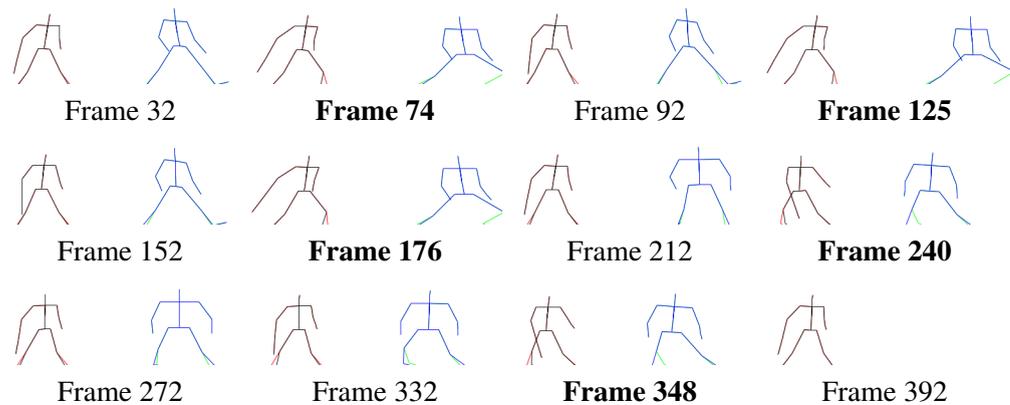
FIGURE D.3: Generated assessment and adjustment for *p04\_inner\_thighs\_worse*. Sequence of the reconstructed skeletons of the user's performance (in red) and its adjustment to the tutor's (in black) with  $\gamma = 0.5$ . Additionally, the synchronised animation of the tutor's demonstration (in green) and its adjustment to the user (in blue) with  $\gamma = 0.5$ . The animation is to be followed from left to right and top-bottom



Generated feedback:

- Frame 16: “*On this occasion, your left knee needs to be bent a little bit more. And your right knee needs to be bent a little bit more. Try ducking lower.*”
- Frame 111: “*This time, once more, your left knee needs to be bent a little bit more. Again, your right knee needs to be bent a little bit more.*”
- Frame 212: “*Now, again, your left knee needs to be bent a little bit more. Once more, your right knee needs to be bent a little bit more. In addition, your left hip needs to be stretched a little bit more. In addition, your right hip needs to be stretched a little bit more. Try moving your left foot further appart from your right foot. In addition, try moving your right foot further appart from your left foot.*”

FIGURE D.4: Generated assessment and adjustment for *p06\_ankles\_worse*. Sequence of the reconstructed skeletons of the user's performance (in red) and its adjustment to the tutor's (in black) with  $\gamma = 0.5$ . Additionally, the synchronised animation of the tutor's demonstration (in green) and its adjustment to the user (in blue) with  $\gamma = 0.5$ . The animation is to be followed from left to right and top-bottom



Generated feedback:

- Frame 74: “Now, your left knee needs to be bent considerably more. You may try keeping your left foot closer to the body mass center.”
- Frame 125: “Now, again, your left knee needs to be bent considerably more.”
- Frame 176: “On this occasion, once more, your left knee needs to be bent considerably more.”
- Frame 240: “On this occasion, your left knee needs to be stretched much more. Keep your left foot further away from your body mass center.”
- Frame 348: “On this occasion, your right knee needs to be bent much more. Again, your left knee needs to be stretched much more. You may try keeping your right foot closer to the body mass center.”

## Appendix E

# Grammars for assessment

194

LISTING E.4: Assessment grammar for *Ankle stretch*

---

```
type$2 & @LEFT_HIP$2 ->F:"Your left hip needs to be stretched a little bit more"  
type$2 & @LEFT_HIP$3 ->F:"Your left hip needs to be stretched much more"  
type$2 & @LEFT_HIP$4 ->F:"Your left hip needs to be stretched considerably more"  
type$2 & (@LEFT_HIP$2 | @LEFT_HIP$3 | @LEFT_HIP$4)->T:"Try leaning your body further over your left hand side"  
  
type$2 & @LEFT_KNEE$2 ->F:"Your left knee needs to be bent a little bit more"  
type$2 & @LEFT_KNEE$3 ->F:"Your left knee needs to be bent much more"  
type$2 & @LEFT_KNEE$4 ->F:"Your left knee needs to be bent considerably more"  
type$2 & (@LEFT_KNEE$2 | @LEFT_KNEE$3 | @LEFT_KNEE$4)->T:"You may try keeping your left foot closer to the body mass center"  
  
type$2 & @RIGHT_HIP$2 ->F:"Your right hip needs to be stretched a little bit more"  
type$2 & @RIGHT_HIP$3 ->F:"Your right hip needs to be stretched much more"
```

```

type$2 & @RIGHT_HIP$4 ->F:"Your right hip needs to be stretched considerably more"

type$2 & @RIGHT_KNEE$2 ->F:"Your right knee needs to be stretched a little bit more"
type$2 & @RIGHT_KNEE$3 ->F:"Your right knee needs to be stretched much more"
type$2 & @RIGHT_KNEE$4 ->F:"Your right knee needs to be stretched considerably more"
type$2 & (@RIGHT_KNEE$2 | @RIGHT_KNEE$3 | @RIGHT_KNEE$4 | @RIGHT_HIP$2 | @RIGHT_HIP$3 | @RIGHT_HIP$4)->T:"Keep your right foot further away
from your body mass center"

type$4 & @RIGHT_HIP$2 ->F:"Your right hip needs to be stretched a little bit more"
type$4 & @RIGHT_HIP$3 ->F:"Your right hip needs to be stretched much more"
type$4 & @RIGHT_HIP$4 ->F:"Your right hip needs to be stretched considerably more"
type$4 & (@RIGHT_HIP$2 | @RIGHT_HIP$3 | @RIGHT_HIP$4)->T:"Try leaning your body further over your right hand side"

type$4 & @RIGHT_KNEE$2 ->F:"Your right knee needs to be bent a little bit more"
type$4 & @RIGHT_KNEE$3 ->F:"Your right knee needs to be bent much more"
type$4 & @RIGHT_KNEE$4 ->F:"Your right knee needs to be bent considerably more"
type$4 & (@RIGHT_KNEE$2 | @RIGHT_KNEE$3 | @RIGHT_KNEE$4)->T:"You may try keeping your right foot closer to the body mass center"

type$4 & @LEFT_HIP$2 ->F:"Your left hip needs to be stretched a little bit more"
type$4 & @LEFT_HIP$3 ->F:"Your left hip needs to be stretched much more"
type$4 & @LEFT_HIP$4 ->F:"Your left hip needs to be stretched considerably more"

type$4 & @LEFT_KNEE$2 ->F:"Your left knee needs to be stretched a little bit more"
type$4 & @LEFT_KNEE$3 ->F:"Your left knee needs to be stretched much more"
type$4 & @LEFT_KNEE$4 ->F:"Your left knee needs to be stretched considerably more"

```

```
type$4 & (@LEFT_KNEE$2 | @LEFT_KNEE$3 | @LEFT_KNEE$4 | @LEFT_HIP$2 | @LEFT_HIP$3 | @LEFT_HIP$4)->T:"Keep your left foot further away from your
body mass center"
```

---

#### LISTING E.5: Assessment grammar for *Arm raises*

---

```
type$2 & @LEFT_SHOULDER$2 ->F:"Your left arm should be raised a little bit higher up"
type$2 & @LEFT_SHOULDER$3 ->F:"Your left arm should be raised much higher up"
type$2 & @LEFT_SHOULDER$4 ->F:"You are not rising your left arm"

type$2 & @RIGHT_SHOULDER$2 ->F:"Your right arm should be raised a little bit higher up"
type$2 & @RIGHT_SHOULDER$3 ->F:"Your right arm should be raised much higher up"
type$2 & @RIGHT_SHOULDER$4 ->F:"You are not rising your right arm"

type$2 & (@LEFT_SHOULDER$2 | @LEFT_SHOULDER$3 | @LEFT_SHOULDER$4 | @RIGHT_SHOULDER$2 | @RIGHT_SHOULDER$3 | @RIGHT_SHOULDER$4) ->T:"Look at the
instructor and try to reach up with your arms on top of your head"

type$2 & @LEFT_ELBOW$2 ->F:"Your left arm should be stretched a little bit more"
type$2 & @LEFT_ELBOW$3 ->F:"Your left arm should be stretched much more"
type$2 & @LEFT_ELBOW$4 ->F:"You are not stretching your left arm"

type$2 & @RIGHT_ELBOW$2 ->F:"Your right arm should be stretched a little bit more"
type$2 & @RIGHT_ELBOW$3 ->F:"Your right arm should be stretched much more"
type$2 & @RIGHT_ELBOW$4 ->F:"You are not stretching your right arm"

type$2 & (@LEFT_ELBOW$2 | @LEFT_ELBOW$3 | @LEFT_ELBOW$4 | @RIGHT_ELBOW$2 | @RIGHT_ELBOW$3 | @RIGHT_ELBOW$4) ->T:"Try to stretch your arms over
your head, with forearm completely parallel to your arm"
```

---

#### LISTING E.6: Assessment grammar for *Calf stretch*

---

```
type$2 & @LEFT_HIP$2 ->F:"Your left hip needs to be stretched a little bit more"
```

```

type$2 & @LEFT_HIP$3 ->F:"Your left hip needs to be stretched much more"
type$2 & @LEFT_HIP$4 ->F:"Your left hip needs to be stretched considerably more"
type$2 & (@LEFT_HIP$2 | @LEFT_HIP$3 | @LEFT_HIP$4)->T:"Try leaning your body further over your left hand side"

type$2 & @LEFT_KNEE$2 ->F:"Your left knee needs to be bent a little bit more"
type$2 & @LEFT_KNEE$3 ->F:"Your left knee needs to be bent much more"
type$2 & @LEFT_KNEE$4 ->F:"Your left knee needs to be bent considerably more"
type$2 & (@LEFT_KNEE$2 | @LEFT_KNEE$3 | @LEFT_KNEE$4)->T:"You may try keeping your left foot closer to the body mass center"

type$4 & @RIGHT_HIP$2 ->F:"Your right hip needs to be stretched a little bit more"
type$4 & @RIGHT_HIP$3 ->F:"Your right hip needs to be stretched much more"
type$4 & @RIGHT_HIP$4 ->F:"Your right hip needs to be stretched considerably more"
type$4 & (@RIGHT_HIP$2 | @RIGHT_HIP$3 | @RIGHT_HIP$4)->T:"Try leaning your body further over your right hand side"

type$4 & @RIGHT_KNEE$2 ->F:"Your right knee needs to be bent a little bit more"
type$4 & @RIGHT_KNEE$3 ->F:"Your right knee needs to be bent much more"
type$4 & @RIGHT_KNEE$4 ->F:"Your right knee needs to be bent considerably more"
type$4 & (@RIGHT_KNEE$2 | @RIGHT_KNEE$3 | @RIGHT_KNEE$4)->T:"You may try keeping your right foot closer to the body mass center"

```

---

**LISTING E.7: Assessment grammar for *Inner thigh stretches***

---

```

type$2 & @LEFT_KNEE$2 ->F:"Your left knee needs to be bent a little bit more"
type$2 & @LEFT_KNEE$3 ->F:"Your left knee needs to be bent much more"
type$2 & @LEFT_KNEE$4 ->F:"Your left knee needs to be bent considerably more"
type$2 & (@LEFT_KNEE$2 | @LEFT_KNEE$3 | @LEFT_KNEE$4)->T:"Try ducking lower"

type$2 & @RIGHT_KNEE$2 ->F:"Your right knee needs to be bent a little bit more"
type$2 & @RIGHT_KNEE$3 ->F:"Your right knee needs to be bent much more"
type$2 & @RIGHT_KNEE$4 ->F:"Your right knee needs to be bent considerably more"
type$2 & (@RIGHT_KNEE$2 | @RIGHT_KNEE$3 | @RIGHT_KNEE$4)->T:"Try ducking lower"

```

```

type$2 & @LEFT_HIP$2 ->F:"Your left hip needs to be stretched a little bit more"
type$2 & @LEFT_HIP$3 ->F:"Your left hip needs to be stretched much more"
type$2 & @LEFT_HIP$4 ->F:"Your left hip needs to be stretched considerably more"
type$2 & (@LEFT_HIP$2 | @LEFT_HIP$3 | @LEFT_HIP$4)->T:"Try moving your left foot further appart from your right foot"

type$2 & @RIGHT_HIP$2 ->F:"Your right hip needs to be stretched a little bit more"
type$2 & @RIGHT_HIP$3 ->F:"Your right hip needs to be stretched much more"
type$2 & @RIGHT_HIP$4 ->F:"Your right hip needs to be stretched considerably more"
type$2 & (@RIGHT_HIP$2 | @RIGHT_HIP$3 | @RIGHT_HIP$4)->T:"Try moving your right foot further appart from your left foot"

type$2 & @SPINE$2 -> F:"You are leaning your body a little bit back"
type$2 & @SPINE$3 -> F:"You are leaning your body back too much, you are at risk of loosing balance"
type$2 & @SPINE$4 -> F:"You are leaning your body forward"
type$2 & (@SPINE$2 | @SPINE$3 | @SPINE$4)->T:"Try to keep your body straight to prevent spine injuries"

```

---

#### LISTING E.8: Assessment grammar for *Shoulder and upper back stretch*

---

```

type$2 & @LEFT_SHOULDER$2 ->F:"Your left arm should be raised a little bit higher up"
type$2 & @LEFT_SHOULDER$3 ->F:"Your left arm should be raised much higher up"
type$2 & @LEFT_SHOULDER$4 ->F:"You are not rising your left arm"

type$2 & @RIGHT_SHOULDER$2 ->F:"Your right arm should be raised a little bit higher up"
type$2 & @RIGHT_SHOULDER$3 ->F:"Your right arm should be raised much higher up"
type$2 & @RIGHT_SHOULDER$4 ->F:"You are not rising your right arm"

type$2 & (!@LEFT_SHOULDER$1 | !@RIGHT_SHOULDER$1) ->T:"Look at the instructor and try to reach up with your arms on top of your head"

type$2 & @LEFT_ELBOW$2 ->F:"Your left arm should be stretched a little bit more"
type$2 & @LEFT_ELBOW$3 ->F:"Your left arm should be stretched much more"

```

type\$2 & @LEFT\_ELBOW\$4 ->F:"You are not stretching your left arm"

type\$2 & @RIGHT\_ELBOW\$2 ->F:"Your right arm should be stretched a little bit more"

type\$2 & @RIGHT\_ELBOW\$3 ->F:"Your right arm should be stretched much more"

type\$2 & @RIGHT\_ELBOW\$4 ->F:"You are not stretching your right arm"

type\$2 & (!@LEFT\_ELBOW\$1 | !@RIGHT\_ELBOW\$1) ->T:"Try to stretch your arms over your head, with forearm completely parallel to your arm"

type\$3 & (@LEFT\_SHOULDER\$2 | @LEFT\_ELBOW\$2) ->F:"Your left arm needs to be stretched a little bit more"

type\$3 & (@LEFT\_SHOULDER\$3 | @LEFT\_ELBOW\$3) ->F:"Your left arm needs to be stretched much more"

type\$3 & (@LEFT\_SHOULDER\$4 | @LEFT\_ELBOW\$4) ->F:"You are not stretching your left arm"

type\$3 & (@RIGHT\_SHOULDER\$2 | @RIGHT\_ELBOW\$2) ->F:"Your right arm needs to be stretched a little bit more"

type\$3 & (@RIGHT\_SHOULDER\$3 | @RIGHT\_ELBOW\$3) ->F:"Your right arm needs to be stretched much more"

type\$3 & (@RIGHT\_SHOULDER\$4 | @RIGHT\_ELBOW\$4) ->F:"You are not stretching your right arm"

type\$3 & (!@LEFT\_SHOULDER\$1 | !@RIGHT\_SHOULDER\$1 | !@LEFT\_ELBOW\$1 | !@RIGHT\_ELBOW\$1) ->T:"Look at the instructor, your upper limbs should be stretched horizontally on a T pose"

---