Movement recognition using context: A lexical approach based on coherence

Movement Recognition using Context: a Lexical Approach Based on Coherence

Alessandra Mileo[1], Stefano Pinardi[1], and Roberto Bisiani[1]

Department of Informatics, Systems and Communication, University of Milan-Bicocca, viale Sarca 336/14, I–20126 Milan

Abstract. Movement recognition constitutes a central task in home-based assisted living environments and in many application domains where activity recognition is crucial. Solutions in these application areas often rely on an heterogeneous collection of body-sensors whose diversity and lack of precision has to be compensated by advanced techniques for feature extraction and analysis. Although there are well established quantitative methods in machine learning for robotics and neighboring fields for addressing these problems, they lack advanced knowledge representation and reasoning capacities that may help understanding through contextualization.
Such capabilities are not only useful in dealing with lacking and imprecise information, but moreover they allow for a better inclusion of semantic information and more general domain-related knowledge.
We address this problem and investigate how a lexical approach to multi-sensor analysis can be combined with answer set programming to support movement recognition. A semantic notion of contextual coherence is formalized and qualitative optimization criteria are introduced in the reasoning process. We report upon a first experimental evaluation of the lexical approach to multi-sensor analysis and discuss the potentials of knowledge-based contextualization of movements in reducing the error rate.

1 Introduction and Motivations

Movement recognition is an important aspect of situation assessment both in Ambient Intelligence and Healthcare applications [1]. It is also important in order to forecast critical situations like a fall or a stroke, to understand emotional patterns from position of the body [2, 3], to track activities [4, 5]. It is important also for gait and posture analysis, human computer interaction, and in motion recognition and capture [6, 7, 4, 8].

It is possible to use video cameras for movement classification, but with a video camera you have to segment body from the background, identify body parts, solve luminance and hidden parts problems, and target the monitored person when more people are present in the area. Video movement analysis is a useful technique in hospital but it can hardly be used in a day by day analysis to classify movements in any natural condition like real sport analysis, healthcare applications, medical or social surveillance protocols [9–11].

It is possible to recognize the movements of a person using wearable inertial sensors, as shown by various studies and applications [12, 13, 1, 4, 5, 14]. Wearable sensors usually contain inertial devices like accelerometers and gyroscopes to detect linear and torque forces: they can be placed on different segments of the body to analyze movements. Some sensors also use magnetometers in order to identify the north-pole direction to determine Euler angles in respect to a fixed reference system [8]. Also, sensors can be connected to a wireless network in order to facilitate the collection of data [13].

Many different ways to use inertial sensors for movement classification have been described in the literature; they are mainly based on Machine Learning techniques [4]. Some researchers use sensors placed on a single spot of the body [1], others use many sensors positioned on different segments of the body [12, 1, 4], others use a multimodal approach using both microphones and inertial sensors [6, 15].

The advantage of using inertial sensors for movement recognition are patent. We know with absolute certainty which segment of the body data come from. We do not have to solve hidden surfaces problems or identify the correct person in a crowded situation. Also inertial sensors can be used in any natural situation and are more respectful of privacy. On the other hand, movement classification using inertial sensors is still an open subject of research and needs to be well understood both in the field of data analysis with machine learning techniques and in the reasoning area.

The rationale is that these methods are useful to create a "lexicon" of movements: the proposed methods have significant error rates, and use small vocabularies [4]. But movements are not only isolated events, they have a "lexical context", are causally and logically connected, and are space dependent.

We are interested in classifying movements with a Machine Learning approach aimed to create a rich user-independent vocabulary, and in exploiting our knowledge of legal sequences of movements as a pattern to reason about movements in order to validate or reject one or more actions in a given scenario.

Our modeling and reasoning technique is based on Answer Set Programming (ASP), a declarative logic programming framework, combining a compact, flexible, and expressive modeling language with high computational performance. The knowledge-based approach provides a more flexible way to reason about semantically-annotated sequences, compared to pure quantitative approaches.

We will use a new proposed Machine Learning method for "lexical analysis" that has a good accuracy 95.23% - 97.63% [12]. Our methodology is inspired by machine learning techniques used for information retrieval and text mining [16], with some adaptation.

On top of this analysis, a further level of knowledge-based validation is in charge of reasoning about meta-patterns of sequences as well as contextual constraints to reduce error rate.

The ASP based reasoning process follows the common generate and test methodology in (i) generating the space of legal patterns according to semantic validation and (ii) exploring the search space by applying efficient solving tech-

niques to compensate for possible errors by enforcing constraint satisfaction and optimization.

The lexical analysis of features for classification of movements is described in Section 2. Section 3 introduces the ASP formalism and presents our modeling and reasoning strategies, while a preliminary evaluation of the potentials of our approach is presented in Section 4. A short discussion follows in Section 5.

2 Movements Classification: from Sensors to Activity

2.1 Sensors and Features Extraction

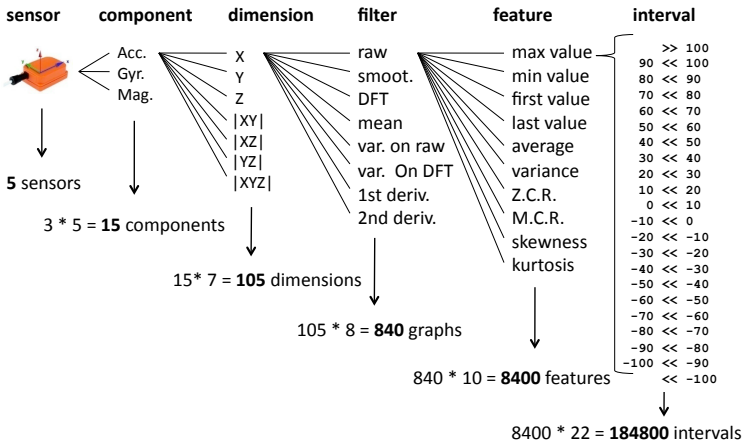
We used five MTx inertial sensors produced by XSens [8], these sensors have three different devices mounted on board: a 3D accelerometer, a 3D Gyroscope, and a 3D magnetometer. Informations are represented as a three dimensional vector in the sensors reference frame providing data about linear forces (ms/sec^2), torque forces (rad/sec) and earth-magnetic field intensity (direction) (milli-Tesla). Even if it is possible to represent vectors in a geo-referenced fixed frame with the MTx sensors, we preferred this configuration because many technologies do not have a fixed frame reference system, and we want to study the problem in the more generic and technologically neutral situation.

In order to create a flexible mechanism to classify movements, we extract very generic features that are not dependent on the application domain. Every sensor component - the accelerometer, gyroscope and magnetometer - returns one information for every spatial dimension (X, Y, Z). Every component information has also been considered in its planar norm representation ($|XY|$, $|XZ|$, $|YZ|$) and in its 3D norm representation ($|XYZ|$), for a total of 7 data per sensor. These data have been filtered using eight functions (null, smoothing, low pass, mean, variance, variance with low pass, first derivative, second derivative) generating 840 transformations of the original data. Then, ten generic features have been chosen (Maximum value, Minimum value, First Sample, Last Sample, Mean, Variance, Zero Crossing Rare, Mean Crossing Rate, Skewness, Kurtosis) for a total number of 8400 features. Finally, features have been quantized roughly into 20 intervals (see Figure1). At the end, every action generates a sparse binary vector of 184800 dimensions. This vector is used to create the classification pattern of movements that constitutes the Lexicon or Dictionary of the application scenario.

2.2 Features Analysis

All features do not have the same relevance depending on population and lexical contexts: some features are more frequent within the population, others can be more or less spread inside the given vocabulary of actions. To transform these qualitative considerations in a quantitative measurement we introduced two weights: the FF (Feature Frequency) and IVFF (Inverse Vocabulary Feature Frequency). Feature Frequency takes into account distributions of the feature per class in the population, as shown in Equation 1:

Fig. 1. Feature extraction process. Actions are transformed in a binary sparse vector of 184.400 values.



$$FF_{ij} = \frac{n_{i,j}}{|P|} \tag{1}$$

where $n_{i,j}$ is the number of occurrences of the feature σ_i in the action a_j and $|P|$ represents the cardinality of the population.

Inverse Vocabulary Feature Frequency weights features according to their discriminatory ability within the dictionary of actions as shown in Equation 2:

$$IVFF_i = log \frac{|A|}{|a : \sigma_i \in a|} \tag{2}$$

where $|A|$ is the cardinality of the dictionary, and $|a : \sigma_i \in a|$ represents the number of actions in which feature σ_i assumes the same value.

Every feature is weighted by multiplying FF and IVF as shown in Equation 3:

$$W_{i,j} = FF_{i,j} * IVFF_i \tag{3}$$

2.3 Vocabulary and Classification

Actions are feature vectors placed into a feature-actions matrix transformed with the weighting operation described in Equation 3. The action to be recognized is

a n-dimensional vector weighted using this transformation. This action is considered a *query* in the feature-action space. Through a set of similarity algorithms the most similar action is calculated, then results are compared with the ground truth. We used three similarity algorithms: Ranking, Cosine Similarity and Euclidean Distance defined, respectively, as follows:

$$rank_j = \sum_{i=1}^{n} W_{i,j} \tag{4}$$

$$dist_i = \sqrt{\sum_{i=1}^{n} (W_{i,j} - q_{i,j})^2} \tag{5}$$

$$cos\theta = \frac{W_{i,j} * q_{i,j}}{|W_{i,j}||q_{i,j}|} \tag{6}$$

where $W_{i,j}$ represents the weight of the σ_i interval of action a_j of the TrainingSet, and $q_{i,j}$ is the IVFF value associated to the feature of the query.

Every action to be classified, hit all features in one of the intervals; every interval for each action in the vocabulary, is associated to a weight for that feature interval in the action, as defined in Equation 3. The rank of an action is given by the sum of weights associated to the intervals of all features interested by the action. The higher the sum of weight, the more similar the action to be classified is to the action in the vocabulary.

We used a Leave One Out Cross Validation (LOOCV) method to test the accuracy of the recognition on two different databases: an internal database NIDA 1.0 with 273 samples, and a public database WARD 1.0 with 1270 samples, obtaining preliminary results illustrated in Section 4.

The accuracy of our method is high when we use five sensors on different parts of the body: we reached an accuracy of 95.23% on NIDA 1.0 and 97.74% on WARD 1.0 outperforming the results presented in the literature on WARD database [12, 13]. There are many situations where it is not possible to wear many sensors on the body for social acceptance or comfort, for example with ill or elder people in health care scenarios. If we use a single sensor accuracy decreases: with a single sensor on the hip the accuracy rate is 81.31%, with only one sensor on the right wrist we have an accuracy of 82.78%, and using just the right ankle we have an accuracy of 83.15%. In these situations the classifier makes errors in almost all the actions (see 3), and the error rate is higher on some specific actions reducing the general performance. We want to improve accuracy reducing the error rate even in the single sensor scenario, and the logic-based contextual inference can help in doing this in a flexible and performant way.

3 Knowledge-Based Support to Movement Recognition

3.1 ASP Basics

We assume the reader to be familiar with the terminology and basic definitions of ASP (see [17] for details). In what follows, we rely on the language sup-

ported by grounders *lparse* [18] and *gringo* [19], providing normal and choice rules, cardinality and integrity constraints, as well as aggregates and optimization statements. As usual, rules with variables are regarded as representatives for all respective ground instances.

3.2 Model of Movement

The logical formalization of our model of movements is provided in terms of three aspects of body motion (referred to as *classes*), each of them characterized by a set of values and one or more additional attributes for that value.

The current[1] list of values and attributes for each class are summarized in Table 1. Please note that attributes related to values of class *posture* indicate where the posture is assumed to be held, while for values of the other classes, the attribute specifies an additional description of how the movement is performed.

Class	**Value**	**Attribute**
posture	sit, stand	{chair, bed}
	lay	{bed}
motion	walk	{forward, upstairs, downstairs, fast}
	jump	{once}
	open	{circular, sliding}
	kick	{frontal, lateral}
heading	right, left	{90, 180}

Table 1. Aspects of body motion included in our vocabulary

Each tuple $< Class, Value, Attribute >$ we define, corresponds to a *word* of the vocabulary of movements introduced in Section 2.

Whenever a new *word* is classified by the underlying mechanism illustrated in Section 2, the knowledge-based representation associates a time step to the logical classification of the action, in a predicate of the form:

$$sensed(Class, Value, Attribute, TimeStep)$$

Extending the vocabulary is a straightforward activity in our model, because it can be done by extending the range of possible attributes of a value, adding new values or adding new classes. The *state* of body motion is determined by three tuples of the form $< Class, Value, Attribute >$, one for each of the three classes, at the same time step T.

[1] In this preliminary analysis we reduced the classes of movements we want to reason about, and their values, in order to better illustrate the reasoning principles via examples.

In order to validate tuples identified by the underlying classification method, we introduce two additional properties for the movement recognition problem: the *semantic distance* measure between to subsequent tuples of the same class, and the *state coherence* of a tuple of a given class, with respect to tuples of the other classes at the same time step. It is worth mentioning that we apply the inertia law on tuples across time steps.

Semantic Distance between two tuples $X =< Class, Value_1, Attribute_1 >$ and $Y =< Class, Value_2, Attribute_2 >$ represented by predicate $dist(X, Y, N)$, is the minimum number of semantically meaningful transitions N that can lead from X to Y.

State Coherence for a tuple $X =< Class, Value, Attribute >$ is the number of (ground) state constraints that are violated at a given time step T by assuming that the identification of X is correct.

Considering our initial vocabulary, semantic distance is mainly concerned with tuples of the form $< posture, V, A >$ because for the other classes, each sequence of values is admitted, therefore we have that the semantic distance is equal to one for every possible sequence of triples where $Class \in \{heading, motion\}$. Distances between triples that are related to class *posture* are summarised in Table 2.

Source Tuple	**Target Tuple**	**Condition**	**Distance**
$< posture, P, V >$	$< posture, P_1, V >$	$P_1 \neq P$	1
$< posture, P, V >$	$< posture, P, V_1 >$	$V_1 \neq V$	2
$< posture, lay, V >$	$< posture, sit, V_1 >$	$V_1 \neq V$	2
$< posture, lay, V >$	$< posture, stand, V_1 >$	$V_1 \neq V$	3
$< posture, sit, V >$	$< posture, X_1, V_1 >$	$(V_1 \neq V) \vee (X_1 \neq sit)$	3

Table 2. Distances between tuples

As for state coherence, we define a set of constraints that are violated for some combination of triples at a given time step. Let us consider our reduced vocabulary of body movements described in Table 1, the state constraints we define express the following concepts:

- a tuple of the form $< posture, sit, A_p >$, for any attribute A_p[2] in the domain of *sit* is not coherent with tuples of the form $< motion, V_i, A_m >$ for any attribute A_m in the domain of the correspondent V_i and $V_i \in \{walk, open\}$;

[2] Note that in our notation, upper-case names refer to variables and have to be instantiated over their domain.

– in a similar way, a tuple of the form $< posture, lay, A_p >$, for any attribute A_p in the domain of *lay* is not coherent with tuples of the form $< motion, V_i, A_i >$ for any attribute A_i in the domain of the correspondent V_i and $V_i \in \{walk, jump, open\}$.

In the next section we illustrate how to reason about contextual coherence to validate actions recognised by the lexicographic classification of movements. Plausible sequences of actions are selected via optimization.

3.3 Movement Recognition: a Contextual View

The contextual validation of activities (or movements) is based on the definition of properties of the activity identified by the underlying classification process. In our solution, we identified two properties whose combination can determine a direct validation or the need for a change in the classification.

Shortest Distance property at a given time step T considers a tuple A_{t-1} that has been validated at time $T-1$ and verifies whether the new tuple A_t to be validated at time T is in the set of possible tuples having distance 1 from A_{t-1}.

State Coherence property at a given time step T for a tuple A_t holds whenever all (100%) of the state constraints are satisfied by the validation of A_t[3].

When the vocabulary is extended or new sensor information is introduced, we can easily re-define or extend the list of properties. An example can be the introduction of localization information in the definition of coherence: to take such information into account, we just have to introduce additional constraints to be satisfied for the *state coherence* property.

As mentioned earlier in this section, a classification represented by a tuple $A_t =< Class, Value, Attribute >$ can be associated to

– a *valid* status, identified by the fact that the logic predicate $valid(A_t)$ holds for A_t;
– a *switched* status, identified by fact that logic predicate $switched(A_t, A_t^{'})$ holds for A_t, given that A_t has been identified at time step T ($sensed(A_t, T)$) but it has been switched to the movement identified by tuple $A_t^{'}$;
– an *incomplete* status, identified by the fact the the logic predicate $incomplete(A_t, A_t^{'})$ holds for A_t, given that A_t has been identified at time step T ($sensed(A_t, T)$) but a classification is missing between $A_t^{'}$ and A_t;

Each classification for A_t can verify one, both or none of the contextual properties used for validation at a given time step T.

[3] Note that this is a simplified version of our formalization. In a more flexible version of the reasoning process we want to consider different level of coherence given by the percentage of constraits that are violated. This would allow to introduce further optimization that will be discussed in a future extended version of this paper.

In the trivial case, $dist(A_{t-1}, A_t, 1)$ holds and A_t is state-coherent: the classification is validated and $valid(A_t)$ becomes true.

Otherwise, we need reasoning to support the validation of the following situations:

1. $dist(A_{t-1}, A_t, 1)$ holds and A_t is not state-coherent;
2. $dist(A_{t-1}, A_t, N)$ holds for $N > 1$ but A_t is state-coherent;
3. $dist(A_{t-1}, A_t, N)$ holds for $N > 1$ and A_t is not state-coherent;

The easiest way to validate the classification of A_t in situation **1.**, is to provide acceptable sets of $switched(B_t, B_t^{'}),\ B_t \neq A_t,\ s.t.\ A_t$ becomes state-coherent.

In the other situations, $dist(A_{t-1}, A_t, N)\ with\ N > 1$, and we have to deal with the following two sub-cases:

a. $\exists\ A_t^{'}$ s.t. $dist(A_{t-1}, A_t^{'}, 1)\ \vee\ A_t^{'}$ is coherent at time T
   $\Rightarrow$ A_t is an erroneous classification and plausible switches $switched(A_t, A_t^{'})$ are derived such that $valid(A_t^{'})$ becomes true;
b. $\nexists\ A_t^{'}$ s.t. $dist(A_{t-1}, A_t^{'}, 1) \vee\ A_t^{'}$ is coherent at time T
   $\Rightarrow$ a classification A^* between A_{t-1} and A_t could be missing, $incomplete(A_t, A_t^{'})$ is derived and we can be in one or both of these scenarios:
   - $A^*,\ A_t$ are made state-coherent by switching appropriate tuples $B_t,\ B \neq A$, such that $valid(A_t)$ becomes derivable
   - an error in the classification of A_t is assumed and state-coherence is computed by switching A_t to $A_t^{'}$ such that $valid(A_t^{'})$ becomes derivable.

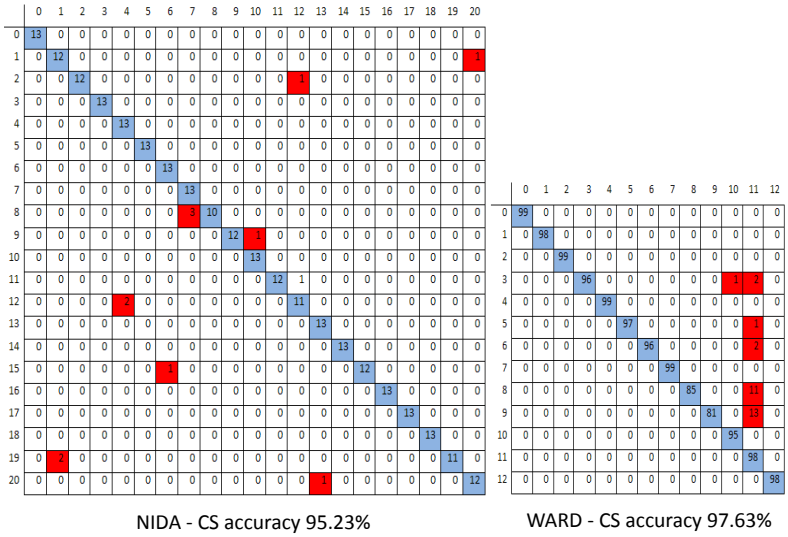
ASP inference is based on a generate-and-test approach. Plausible classifications for an action are generated using a *cardinality constraint* and then checked for *state coherence* and *shortest distance*. If the classification provided by the underlying mechanism described in Section 2 is among them, the system validates it. Otherwise, a different solution is proposed by switching one or more of the movements in the sequence, or by supposing that the sequence is incomplete and that some classifications are missing, or both, resulting in a lot of possibilities. In a similar scenario, default reasoning, non-determinism, choice rules, constraints and optimization via preferences plays a key role in devising effective deduction strategies. For lack of space, we cannot illustrate how all these constructs are used in the ASP encoding. A simple optimization criteria is based on the global minimization of switched tuples, i.e. we prefer to assume incomplete sequences rather than wrong classifications when this lead to a solution, but we can easily change our preference in a declarative way.

4 Evaluation Phase

In this section we illustrate results of preliminary tests on the identification of movements as non contextualized *words* of a *body lexicon*. Althought our machine learning process gives acceptable results, once we reduce the number of sensors, the error rate increases and misleading classifications cannot be identified. We believe that the knowledge-based contextualization of sequences of movements described in Section 3 can help reducing misinterpretations, although we need further testing to estimate percentual reduction of error rate.

4.1 Test Methods

Fig. 2. Confusion Matrix for Cosine Similarity (CS) with five sensors on NIDA and WARD databases.



At first we created a Test-Set of twenty-one different actions called NIDA 1.0. (Nomadis Internal Database of Actions). The NIDA 1.0 database contains movements acquired by the NOMADIS Laboratory of the University of Milano-Bicocca. These acquisitions have been obtained using 5 MTx sensors positioned on the pelvis, on the right and left wrist, and on the right and left ankle. NIDA includes 21 types of actions performed by 7 people (5 male and 2 female) ranging from 19 to 44-years-old, for a total of 273 actions. The complete list of actions is the following:

1. Get up from bed. 2. Get up from a chair. 3. Open a wardrobe. 4. Open a door. 5. Fall. 6. Walk forward 7. Run. 8. Turn left 180 degrees. 9. Turn right 180 degrees. 10. Turn left 90 degrees. 11. Turn right 180 degrees. 12. Karate frontal kick. 13. Karate side kick. 14. Karate punch. 15. Go upstairs. 16. Go downstairs. 17. Jump. 18. Write. 19. Lie down on a bed. 20. Sitting on a chair 21. Heavily sitting on a chair.

We also tested our methodology on a public database called WARD 1.0 (Wearable Action Recognition Database) created at UC-Berkeley [12, 13]. Acquisitions have been obtained positioning 5 sensors on the pelvis, on the right and left wrist, and on the right and left ankle. Each sensor contains a 3-axial

accelerometer and a 2-axial gyroscope; magnetometers are not present. WARD contains 13 types of actions performed by 20 people (7 women and 13 men) ranging from 20 to 79-years-old with 5 repetition per action, for a total o 1200 actions. The list of actions can be found in [13].

4.2 Preliminary Analysis

We used a Leave One Out Cross Validation (LOOCV) method to test the accuracy of the proposed method. The accuracy of the algorithms for the NIDA database are: Ranking 89.74%, Euclidean Distance 95.23%, Cosine similarity 95.23%. The accuracy of algorithms using the WARD database are: Ranking 97.5%, Euclidean Distance 97.74%, Cosine 97.63% . We show the results of the Cosine Similarity also in a synoptic way using a Confusion Matrix (see Figure 2). In the columns the ground truth, in the rows the output of our classification algorithm. Label definitions are given in the above paragraph.

Fig. 3. Confusion Matrix for Cosine Similarity on the NIDA database. Accuracy: 81.31% using only one sensor on the hip.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
2	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	10	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
6	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	9	2	2	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	4	8	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	2	0	10	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	2	1	10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	1	0	1	0	0	0	0	8	2	1	0	0	0	0	0	0	0
12	0	0	0	0	2	0	0	0	0	0	0	1	8	2	0	0	0	0	0	0	0
13	0	0	0	0	1	0	1	0	0	0	0	2	1	8	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	3	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	10	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0
19	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0
20	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	10

5 Discussion

We presented a hybrid approach to movement recognition by combining and extending standard quantitative methods in a knowledge-based yet qualitative framework. To this end, we took advantage of the knowledge representation and reasoning capacities of ASP for providing semantic contextualization.

Although we have not discussed the encoding in detail, it allows for easy customization and extensibility to a richer “lexicon” of movements. All in all, the contextual support of the high-level ASP specification makes the major difference of our approach to potential alternatives, and it seems hard to envisage in a purely quantitative settings. This preliminary work is only a starting point.

Future work will have to address more extensive and systematic experiments in various simulated as well as real scenarios. The problem of segmentation of a sequence of movements needs to be taken into account in order to evaluate the true scalability of our approach.

References

1. Merico, D., Mileo, A., Pinardi, S., Bisiani, R.: A Logical Approach to Home Healthcare with Intelligent Sensor-Network Support. The Computer Journal (2009) bxn074
2. Kleinsmith, A., Bianchi-Berthouze, N.: Recognizing affective dimensions from body posture. In: ACII. (2007) 48–58
3. Castellano, G., Villalba, S.D., Camurri, A.: Recognising human emotions from body movement and gesture dynamics. In: ACII. (2007) 71–82
4. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Pervasive. (2004) 1–17
5. woo Lee, S., Mase, K.: Activity and location recognition using wearable sensors. IEEE Pervasive Computing **1** (2002) 24–32
6. Lester, J., Choudhury, T., Borriello, G.: A practical approach to recognizing physical activities. In: Pervasive. (2006) 1–16
7. Dessere, E., Legrand, L.: First results of a complete marker-free methodology for human gait analysis. In: IEEE EMBC 2005. (2005)
8. (http//www.xsens.com/)
9. Cameron, J., Lasenby, J.: Estimating human skeleton parameters and configuration in real-time from markered optical motion capture. In: AMDO. (2008) 92–101
10. Okada, R., Stenger, B.: A single camera motion capture system for human-computer interaction. IEICE Transactions **91-D**(7) (2008) 1855–1862
11. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding **81**(3) (2001) 231–268
12. Pinardi, S., B.R.: Movement recognition with intelligent multisensor analysis, a lexical approach. (2010) To appear.
13. Yang, A.Y., Jafari, R., Sastry, S.S., Bajcsy, R.: Distributed recognition of human actions using wearable motion sensor networks. Ambient Intelligence and Smart Environments (2009) To appear.
14. Mntyjrvi J, Himberg J, S.T.: Recognizing human motion with multiple acceleration sensors. (2001) 747–752
15. Lester, J., Choudhury, T., Kern, N., Borriello, G., Hannaford, B.: A hybrid discriminative/generative approach for modeling human activities. In: In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI. (2005) 766–772
16. Gerard Salton, A.W., Yang, C.: A vector space model for information retrieval. Journal of the American Society for Information Science **18**(11) (1975) 613–620
17. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
18. Syrjänen, T.: Lparse 1.0 user's manual. (http://www.tcs.hut.fi/Software/smodels/lparse.ps.gz)
19. Gebser, M., Schaub, T., Thiele, S.: Gringo : A new grounder for answer set programming. In: LPNMR. (2007) 266–271