Learning Parallel Grammar Systems for a Human Activity Language

Gutemberg Guerra-Filho and Yiannis Aloimonos
Computer Vision Laboratory
Department of Computer Science
University of Maryland, College Park 20742 USA
guerra@cs.umd.edu, yiannis@cfar.umd.edu

Abstract

We have empirically discovered that the space of human actions has a linguistic structure. This is a sensory-motor space consisting of the evolution of the joint angles of the human body in movement. The space of human activity has its own phonemes, morphemes, and sentences. In kinetology, the phonology of human movement, we define atomic segments (kinetemes) that are used to compose human activity. In this paper, we present a morphological representation that explicitly contains the subset of actuators responsible for the activity, the synchronization rules modeling coordination among these actuators, and the motion pattern performed by each participating actuator. We model a human action with a novel formal grammar system, named Parallel Synchronous Grammar System (PSGS), adapted from Parallel Communicating Grammar Systems (PCGS). We propose a heuristic PArallel Learning (PAL) algorithm for the automatic inference of a PSGS. Our algorithm is used in the learning of human activity. Instead of a sequence of sentences, the input is a single string for each actuator in the body. The algorithm infers the components of the grammar system as a subset of actuators, a CFG grammar for the language of each component, and synchronization rules. Our framework is evaluated with synthetic data and real motion data from a large scale motion capture database containing around 200 different actions corresponding to verbs associated with voluntary observable movement. On synthetic data, our algorithm achieves 100% success rate with a noise level up to 7%.

1. Introduction

Human movement is a natural phenomenon involving a number of independent actuators: articulated body parts or joint angles. The actuators coordinate their actions to achieve some specific common purpose. In human motion modeling, the actuators consist of a fixed set ranging from total body to a single joint. This assumption neglects the independent behavior of the actuators over different activities. Furthermore, an approach modeling explicitly the variability of the set of actuators is more robust concerning occlusion and field of view limitations in the observation process.

The different strategies of parallel and synchronous interaction among actuators play an important role in human movement. Therefore, a movement representation for a specific human activity should include the set of parallel actuators involved in the activity, the synchronization rules among these actuators, and the motion associated with each participating actuator.

Human activity representation involves several challenging problems and has many applications in different areas. In Robotics, adequate movement models are detailed domain knowledge of the solution for complex nonlinear dynamics problems related to motor coordination. The representations make these problems highly structured and suited for path planning of motor control towards skill acquisition in humanoid robots. In Computer Vision, surveillance is achieved with automatic activity detection and recognition based on action representations. They also assist video annotation with efficient storage, transmission, editing, browsing, indexing, and retrieval of the motion data in visual media. In Kinesiology, athletic performance analysis optimizes the training process and improves performance. In Biomechanics, rehabilitation medicine detects, describes anomalies, and helps in the development of treatments. In Performing Arts, motion representations interface with dance notation systems. In Computer Graphics, computer animation performs realistic motion synthesis and composition. Additional fields include human-computer interaction, virtual reality, and augmented reality.

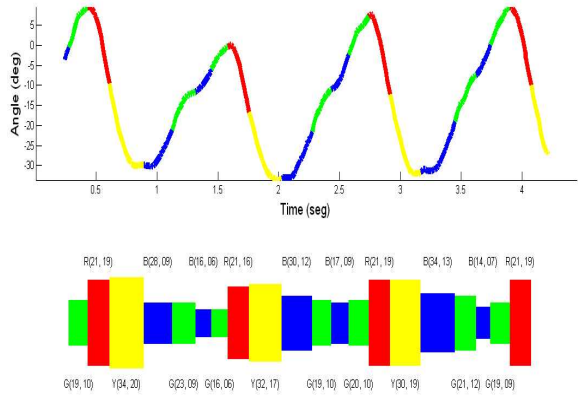
We propose a linguistic framework for the modeling and learning of human activity representations. Our ultimate goal is to discover a sensory-motor language, denoted as Human Activity Language (HAL), which represents the sequential and parallel aspects of human movement with perceptual and generational properties. Our approach aims to find a linguistic structure for human movement with analogs of phonology, morphology, and syntax.

The availability of a language characterizing human action has implications with regards to the grounding problem, to the universal grammar theory, to the origin of human language and its acquisition process. Besides these

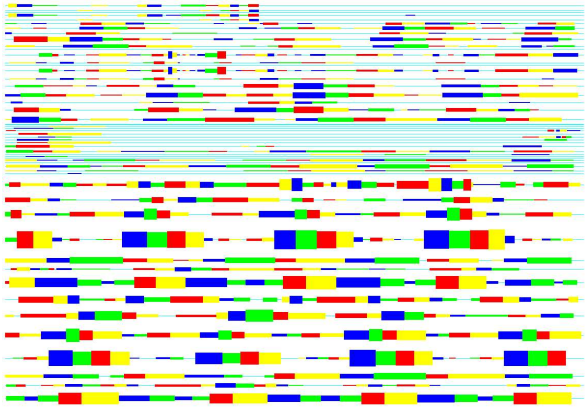
theoretical issues, a linguistic representation for human activity has several practical advantages. A compact specification for human activity leads to compression and better efficiency. Once a symbolic linguistic representation is provided, natural language processing and speech recognition are sources of methods that could be applied to activity understanding. A non-arbitrary symbolic representation allows the use of techniques of symbolic reasoning for inference and other cognitive tasks (e.g. recognition) on human activities. This framework could also be used as a basic module of a symbolic query language for the processing of multimedia data.

In this paper, we discuss the morphological part of our linguistic framework where we present the steps required for the construction of a praxicon, a human activity lexicon, through the learning of grammar systems for human actions. The discovery of HAL involves learning the syntax of human motion and requires the construction of a praxicon. The morphology assumes a non-arbitrary symbolic representation of the human movement. In order to analyze the morphology of a particular action, we are given a symbolic representation for the motion of each actuator associated with several repeated performances of this action. This representation originates from kinetology [9], the phonology of human movement. A kinetological system consists in the identification (segmentation) and representation of atomic motion according to five evaluation principles: compactness, view-invariance, reproducibility, selectivity, and reconstructivity.

In order to segment human movement, we consider each actuator independently. An actuator is associated with a joint angle specifying the original 3D motion of the actuator. The segmentation process assigns one state to each instant of the movement for the actuator in consideration. Contiguous instants assigned to the same state belong to the same segment. We define a state according to the sign of derivatives of a joint angle function (see Fig. 1a). The derivatives used in our segmentation are velocity (first derivative) and acceleration (second derivative). Each segment corresponds to an atom α, where $\alpha \in \{\mathbf{B}, \mathbf{G}, \mathbf{R}, \mathbf{Y}\}$ is a symbol associated with the segment's state. The input for our parallel learning algorithm is a string of symbols for each actuator in the body (see Fig. 1b). Segments with the same state are clustered into classes associated with these symbols. This set of strings for the whole body defines a single structure denoted as *actiongram*. An actiongram A has n strings $A_1, \ldots, A_n$. Each string A_i contains a (possibly different) number of m_i symbols. Each symbol $A_i(j)$ is associated with the time period of the corresponding segment.



(a) Joint angle function segmentation.



(b) Actiongram.

Fig. 1: A compact symbolic representation for human movement.

The inspiration for a sensory-motor linguistic approach to human activity representation comes from the evidences in cognitive sciences, neurophysiology, and psychophysics. The existence of mirror neurons [5] in humans suggests that the same representation for motor information related to body movement is also used in the brain for perceptual tasks. Motor tapes [11] are explicit representations of a movement trajectory in memory. When an agent needs information on how to perform an action, it finds the appropriate template in memory and executes it. A linguistic framework for a common representation is a reasonable approach since there is evidence that language is semantically grounded also in action [6, 17].

Given sequences A_i of symbols associated with motor primitives representing the movement for all actuators i when a specific activity is performed repeated times. The problem addressed in this paper is to identify the set C of true actuators responsible for the specific goal achieved with this activity, to learn the motion structure for all actuators in C, and the synchronization rules among these actuators. A praxicon is built by solving this problem for all actions in a large lexicon of verbs associated with observable human movement [8]. Although the input

concerns a specific action performed several times, we aim to model any general activity, not only restricted to repetitive movement.

We pose this problem as the grammatical inference of a novel grammar system modeling the human activity. As a formal model, we propose a Parallel Synchronous Grammar System where each component grammar corresponds to an actuator. We present a novel heuristic parallel learning algorithm to induce this grammar system. Our algorithm does not assume knowledge of either the number of components or the language components of the grammar system being inferred. The input is a single symbolic stream (string) per actuator instead of a sequence of sentences.

The results of our approach are both theoretical, concerning the heuristic inference of a parallel grammar system, and empirical, in terms of human movement representation and learning. An advantage of parallel learning over plain sequential learning is that problems with overgeneralization are resolved in parallel learning. Sequential learning is able to infer the structure of a single sequence of symbols. This structure corresponds to a forest of binary trees, where each node in a tree is associated with a grammar rule in a normal form. A sequential learning algorithm may keep merging adjacent root nodes into single rules (trees) and, consequently, overgeneralization happens when unrelated rules are generalized. In parallel learning, we use the learned synchronized rules to resolve overgeneralization. This way, root nodes are merged only if the new rule is synchronized with other rules in different components of the grammar system. This way, overgeneralization is avoided since synchronization guarantees a relationship between the merged rules.

We evaluated our approach with synthetic data and real human motion data. We created synthetic actiongrams and tested our method with increasing levels of noise. The algorithm achieved 100% success with a noise level up to 7%. The experimental validation of our linguistic framework is performed in a motion capture database as well. Our motion capture database contains around 200 different actions corresponding to verbs associated with voluntary observable movement. The actions are not limited to any specific domain. Instead, the database includes actions of several types: manipulative (prehension and dexterity), non-locomotor, locomotor, and interaction.

The rest of this paper is organized as follows. In Section 2, we review related work in human activity representation, grammatical inference, and grammar systems. Our parallel learning algorithm to infer a Parallel Synchronous Grammar System is presented and evaluated in Section 3. In Section 4, we discuss kinetology and demonstrate our approach on human activity learning.

2. Related Work

The work discussed in this paper is based on research developed in three main areas: human action representation, grammatical inference, and grammar systems. We use the formalization of grammar systems and present an induction algorithm for the learning of human activity.

2.1. Human Activity Representation

Stuart and Bradley [23] find interpolation sequences between pairs of body postures using A^* search in a set of transition graphs built from corpora of human movement. These graphs capture the progressions of a single joint in the corpus.

Hidden Markov Models (HMM) are vastly used to characterize movement sequences [24]. Alon at al. [1] estimate a finite mixture of HMMs using an Expectation Maximization formulation. In this approach, segments are partially assigned to all clusters corresponding to HMMs. Brand and Hertzmann [2] extend HMM with a multidimensional style variable used to vary its parameters. They learn motion patterns from a set of motion sequences. HMMs are essentially probabilistic finite state automata. In this sense, a stochastic context-free grammar (SCFG) is a generalized model, which relaxes some structural limitations. Ivanov and Bobick [12] use a single SCFG to parse activities and interactions between multiple agents.

Sidenbladh at al. [21] construct a low dimensional linear model of the human motion. They use Principal Component Analysis (PCA) to reduce the dimensionality of the time series of joint angles. The movement data is structured into a binary tree using the coefficients with larger variance in higher levels of the tree. Jenkins and Matarić [13] use dimensionality reduction to extract motion primitives (spatio-temporal structure) with an extension of the Isomap algorithm. The algorithm performs eigenvalue decomposition on a similarity matrix computed as a geodesic distance between each data pair.

Wang at al. [25] present a segmentation which uses the local minima of velocity and local maxima of change in direction. The segments are hierarchically clustered into classes associated with symbols. A small lexicon is inferred from the symbolic sequence through a language acquisition approach. The lexicon is induced for a single movement stream/string and, consequently, involves only sequential learning which suffers from the overgeneralization problem.

Mörchen at al. [15] present a framework to discover movement patterns from EMG and kinematic measurements represented as multivariate time series. The kinematic time series are reduced to primitive patterns by

manual clustering with Emergent Self-Organizing Maps and no time information. The same consecutive primitives are merged into intervals corresponding to symbolic states. They assume all actuators are participating equally in the action. While they consider all aspects of movement at the same time (total body movement) to find coincident intervals, our approach identifies the relevant actuators involved in the movement automatically and considers actuators independently. Furthermore, in their approach, the pattern events discovered are sparse and cannot be used for the reconstruction of the movement.

To the best of our knowledge, no approach modeling human motion learns the set of actuators involved in an action. Usually, they consider a fixed set of actuators and, since our method induces the appropriate actuator set for each action, a comparison between our technique and others is unfeasible.

2.2. Grammatical Inference

Grammatical inference concerns the induction of the syntax (or the grammar) of a language from a set of labeled sentences. The grammar inference consists in learning a set of rules for generating and recognizing the valid strings that belong to the language. The target grammar is usually modeled as grammars that belong to the Chomsky hierarchy of formal grammars. The literature is vast on methods for learning regular grammars, context free grammars, and stochastic variations [18].

Regular grammars and context free grammars cannot be induced only from positive examples [7]. However, several heuristic techniques learn approximations to the target grammar. The SNPR algorithm [26] learns syntagmatic elements (sequences) and paradigmatic elements (sets) from minimal elements which are perceptual primitives (e.g. letters or phonemes). Each element corresponds to a rule in the learned grammar. The learning involves the concatenation of the most frequent pair of contiguous elements.

Sequitur [16] is an algorithm that infers a hierarchical structure from a sequence of discrete symbols. Sequitur infers a grammar, where each repeated subsequence gives rise to a rule and is replaced by a non-terminal symbol. The algorithm constrains the grammar with two properties: digram uniqueness and rule utility. The algorithm operates by enforcing these constraints on an online stream.

2.3. Grammar Systems

Variants of the classical models in Formal Language Theory are used to specify non-determinism in computing devices with notions such as distribution, parallelism, concurrency, and communication. A *grammar system* consists of several grammars (components) that work together generating a common symbolic state represented by a finite set of strings. The components of the system change the state through rewriting and communication.

We use grammar systems as a formal model to learn the morphological structure of human actions. Other formalizations in Natural Language Processing (e.g. synchronous grammars) are not appropriate. They are used in machine translation to correspond structures in different languages that have the same meaning. In human motion modeling, different actuators play different roles executing synchronously distinct unrelated motor programs.

The most important models of grammar systems are cooperating and parallel grammars. *Cooperating Distributed Grammar Systems* (CDGS) have components working sequentially [3, 14]. Only one component is active at any moment. Therefore, the components take turns in rewriting a common sentential form according to a certain cooperation protocol. *Colonies* are a simplification of CDGS where the components are regular grammars generating finite languages. Sosík and Štýbnar [22] train a Neural Pushdown Deterministic Automaton (NPDA) with sequential access to a set of positive and negative sequences in some language. The NPDA model requires preliminary information about the expected size of the inferred grammar, since the topology of the NPDA does not change during the training. They extract a colony from the trained NPDA with a heuristic algorithm after a hierarchical clustering in the space of neuron states.

A *Parallel Communicating Grammar System* (PCGS) consists of several grammar components working simultaneously in a synchronized manner [20]. The component grammars rewrite their own sentential forms in parallel. They communicate by exchanging their current sentential forms among each other. The requested string becomes part of the sentential form of the receiving grammar. In a returning mode, after sending their partial solutions to others, the components are reset to their axioms and start a new computation. The language generated by the system is the language generated by a distinguished component of the system (master grammar) with the help of the others.

The assumption that communication takes a single step and components continue computation without waiting for the end of communication is not reasonable. Fernau [4] discusses a variant of PCGS with terminal transmission and right-linear components. In this model, the communication is constrained only to the transmission of terminal strings. Therefore, queried components have only terminal strings as sentential forms by definition. An inference algorithm for this model is proposed which uses additional structural information about communication (sentences with query symbols) and the component

languages are learned separately with special care for the master component.

3. Parallel Learning Algorithm

In human movement, we are interested only in the simultaneous synchronized work of the components. The communication feature is unnecessary because it is implicit in motion coordination. We propose a simplified grammar system where strings generated by components are not shared through communication steps. The formal model suggested is a PCGS with rule synchronization [19] and no query symbols. The synchronization among rules in different components is controlled with a set of tuples of rules, possibly one rule for each component, where rules in a tuple are derived simultaneously. We specify the definitions related to our adapted PCGS model below. We assume the reader is familiar with the fundamentals of formal language theory. For further information in formal language theory, the reader is directed to [10].

A *Parallel Synchronous Grammar System* (PSGS) with $n \geq 1$ components is an $(n+3)$-tuple $\Gamma = (N, T, G_1, G_2, \ldots, G_n, M)$, where N is a set of non-terminals and T is a terminal alphabet (N and T are mutually disjoint); $G_i = (N, T, P_i, S_i)$, $1 \leq i \leq n$, are Chomsky grammars with a finite set of production rules P_i over $(N \cup T)$ and a start symbol (axiom) $S_i \in N$; and M is a subset of $(P_1 \cup \{\#\}) \times \ldots \times (P_n \cup \{\#\})$, where $\# \notin (\mathrm{N} \cup \mathrm{T})$ is an additional symbol.

A configuration n-tuple $(x_1, \ldots, x_n)$ of Γ directly derives $(y_1, \ldots, y_n)$, where $x_i, y_i \in (N \cup T)^*$, if we have a direct derivation $x_i \Rightarrow y_i$ in each grammar G_i with x_i not terminal or $x_i = y_i$ when $x_i \in T^*$. Each component uses one of its rewriting rules except those grammars which have already produced a terminal string. At a derivation step, a transition n-tuple $(p_1, \ldots, p_n)$ of M is applied, that is $x_i \Rightarrow y_i$ by the rule p_i, if $p_i \in P_i$, and $x_i = y_i$, if $p_i = \#$. A derivation starts from the initial configuration consisting of the axioms $(S_1, \ldots, S_n)$. The language generated by Γ is $L(\Gamma) = \{(\alpha_1, \ldots, \alpha_n), \alpha_i \in T^* \mid (S_1, \ldots, S_n) \Rightarrow^* (\alpha_1, \ldots, \alpha_n)\}$.

A simple example of a PSGS with four components is $\Gamma = (\{S_1, S_2, S_3, S_4, N_1, \ldots, N_{23}\}, \{a, b, c, d\}, G_1, G_2, G_3, G_4, M)$, where

$P_1 = \{S_1 \rightarrow N_{13}S_1, S_1 \rightarrow N_{13}, N_5 \rightarrow \mathrm{bc}, N_9 \rightarrow \mathrm{a}N_5,$
$N_{10} \rightarrow N_9\mathrm{d}, N_{11} \rightarrow N_{10}N_5, N_{12} \rightarrow N_{11}\mathrm{a},$
$N_{13} \rightarrow N_{12}\mathrm{d}\},$

$P_2 = \{S_2 \rightarrow N_{18}S_2, S_2 \rightarrow N_{18}, N_1 \rightarrow \mathrm{bc}, N_{14} \rightarrow N_1\mathrm{a},$
$N_{15} \rightarrow N_{14}\mathrm{d}, N_{16} \rightarrow N_{15}\mathrm{a}, N_{17} \rightarrow N_{16}N_1,$
$N_{18} \rightarrow N_{17}\mathrm{d}\},$

$P_3 = \{S_3 \rightarrow N_7S_3, S_3 \rightarrow N_7, N_2 \rightarrow \mathrm{cd}, N_3 \rightarrow N_2\mathrm{a},$
$N_4 \rightarrow N_3\mathrm{b}, N_7 \rightarrow N_4N_4\},$

$P_4 = \{S_4 \rightarrow N_{23}S_4, S_4 \rightarrow N_{23}, N_6 \rightarrow \mathrm{bc}, N_{17} \rightarrow \mathrm{a}N_6,$
$N_{20} \rightarrow N_{19}\mathrm{d}, N_{21} \rightarrow N_{20}N_6, N_{22} \rightarrow N_{21}\mathrm{a},$
$N_{23} \rightarrow N_{22}\mathrm{d}\},$ and

$M = \{(S_1 \rightarrow N_{13}S_1, S_2 \rightarrow N_{18}S_2, S_3 \rightarrow N_7S_3, S_4 \rightarrow N_{23}S_4),$
$(S_1 \rightarrow N_{13}, S_2 \rightarrow N_{18}, S_3 \rightarrow N_7, S_4 \rightarrow N_{23}),$
$(N_5 \rightarrow \mathrm{bc}, N_1 \rightarrow \mathrm{bc}, N_4 \rightarrow N_3\mathrm{b}, N_6 \rightarrow \mathrm{bc}),$
$(N_9 \rightarrow \mathrm{a}N_5, N_{14} \rightarrow N_1\mathrm{a}, \#, N_{19} \rightarrow \mathrm{a}N_6),$
$(N_{10} \rightarrow N_9\mathrm{d}, N_{15} \rightarrow N_{14}\mathrm{d}, \#, N_{20} \rightarrow N_{19}\mathrm{d}),$
$(N_{11} \rightarrow N_{10}N_5, N_{16} \rightarrow N_{15}\mathrm{a}, \#, N_{21} \rightarrow N_{20}N_6),$
$(N_{12} \rightarrow N_{11}\mathrm{a}, N_{17} \rightarrow N_{16}N_1, \#, N_{22} \rightarrow N_{21}\mathrm{a}),$
$(N_{13} \rightarrow N_{12}\mathrm{d}, N_{18} \rightarrow N_{17}\mathrm{d}, N_7 \rightarrow N_4N_4,$
$N_{23} \rightarrow N_{22}\mathrm{d})\}.$

An example derivation in Γ is $(S_1, S_2, S_3, S_4) \Rightarrow (N_{13}, N_{18}, N_7, N_{23}) \Rightarrow (N_{12}\mathrm{d}, N_{17}\mathrm{d}, N_4N_4, N_{22}\mathrm{d}) \Rightarrow (N_{11}\mathrm{ad}, N_{16}N_1\mathrm{d}, N_4N_4, N_{21}\mathrm{ad}) \Rightarrow (N_{10}N_5\mathrm{ad}, N_{15}\mathrm{a}N_1\mathrm{d}, N_4N_4, N_{20}N_6\mathrm{ad}) \Rightarrow (N_9\mathrm{d}N_5\mathrm{ad}, N_{14}\mathrm{da}N_1\mathrm{d}, N_4N_4, N_{19}\mathrm{d}N_6\mathrm{ad}) \Rightarrow (\mathrm{a}N_5\mathrm{d}N_5\mathrm{ad}, N_1\mathrm{ada}N_1\mathrm{d}, N_4N_4, \mathrm{a}N_6\mathrm{d}N_6\mathrm{ad}) \Rightarrow (\mathrm{abcdbcad}, \mathrm{bcadabcd}, N_3\mathrm{b}N_3\mathrm{b}, \mathrm{abcdbcad}) \Rightarrow (\mathrm{abcdbcad}, \mathrm{bcadabcd}, N_2\mathrm{ab}N_2\mathrm{ab}, \mathrm{abcdbcad}) \Rightarrow (\mathrm{abcdbcad}, \mathrm{bcadabcd}, \mathrm{cdabcdab}, \mathrm{abcdbcad})$. The corresponding parse trees displaying the structure of this set of strings are shown in Figure 2.

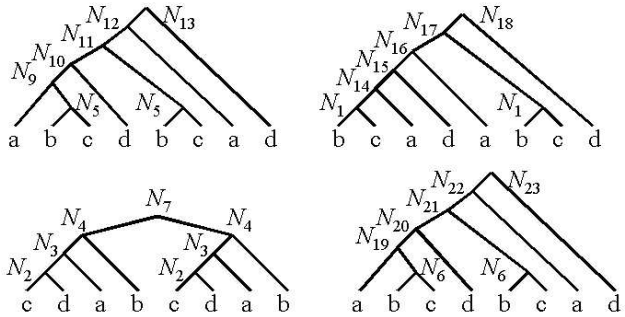


Fig. 2: Parse trees for a Parallel Synchronous Grammar System.

A PSGS consists in a set of CFGs related by synchronized rules. This grammar models a system with a set `A` of different strings $\mathtt{A}_i$ occurring at the same time. Each string $\mathtt{A}_i$ corresponds to the language which will be inferred for a component G_i. Each symbol $\mathtt{A}_i\mathtt{(j)}$ in a string corresponds to a pair $(\mathtt{T}_i\mathtt{(j)}, \mathtt{D}_i\mathtt{(j)})$ for $i = 1, \ldots, \mathtt{m}_i$, where $\mathtt{m}_i$ is the number of symbols in the string $\mathtt{A}_i$. $\mathtt{T}_i\mathtt{(j)}$ is the start time and $\mathtt{D}_i\mathtt{(j)}$ is the time length of the segment corresponding to $\mathtt{A}_i\mathtt{(j)}$. Note that $\mathtt{A}_i\mathtt{(j)} \neq \mathtt{A}_i\mathtt{(j+1)}$ and $\mathtt{T}_i\mathtt{(j)} + \mathtt{D}_i\mathtt{(j)} = \mathtt{T}_i\mathtt{(j+1)}$.

Our PArallel Learning algorithm (see Fig. 3), named PAL, computes the digram frequency within each string independently. The function `DigramFrequency` finds a matrix `df`, where each element $\mathtt{df(i, j)}$ is the number of occurrences of digram $\mathtt{A}_i\mathtt{(j)}\ \mathtt{A}_i\mathtt{(j+1)}$ in string $\mathtt{A}_i$.

```
Algorithm PAL(A, T, D)
 df ← DigramFrequency(A);
 while (∃i | m_i > 1 and max(df) > 1)
  (i, j) ← argmax(df);
  P_i ← P_i ∪ [N_c → A_i(j) A_i(j+1)];
  ReverseRewrite(A, c, i, j);
  R ← SynchronizedRules(A,T,D,R,c,i);
  df ← DigramFrequency(A);
 end

Function SynchronizedRules(A,T,D,R,c,i)
 E_c ← FindOccurrences(A_i, N_c);
 for k = 1, …, c−1
  if (i ≠ q, where N_c ∈ A_i and N_k ∈ A_q)
   E_k ← FindOccurrences(A_q, N_k);
   for u = 1, …, |E_c|; v = 1, …, |E_k|
    if (E_c(u) ∩ E_k(v))
     I(u, v) ← 1;
    end
   end
   if (one-to-one(I))
    R ← R ∪ (N_c, N_k);
   end
  end
 end
```

Fig. 3: Parallel Learning algorithm.

A new rule is created for the digram *d* corresponding to element `(i, j)` with the current maximum frequency. A non-terminal N_c corresponding to a rule $[N_c \rightarrow A_i(j)\ A_i(j+1)]$ is inserted in the set of rules P_i. The procedure `ReverseRewrite` replaces each occurrence of the digram *d* in string A_i with the non-terminal N_c. The new non-terminal is associated with the time period corresponding to the union of the periods of both symbols $A_i(j)$ and $A_i(j+1)$.

The non-terminal N_c is checked for possible synchronized rules with non-terminals in the CFGs of other strings ($i \neq q$). Synchronization between two non-terminals (N_c and N_k) of different CFGs requires these non-terminals to have an intersecting time period ($E_c(u) \cap E_k(v)$) in the different strings generated by their respective CFGs. Synchronization relating two non-terminals in different CFGs is issued if there is a one-to-one mapping (`one-to-one(I)`) of their occurrences in the associated strings. Furthermore, any two mapped occurrences must have intersecting time periods. The function `SynchronizedRules` performs this search for synchronization and incrementally creates a relation `R`, where each pair in this relation represents two synchronized rules in different component grammars. The synchronous tuples in *M* are trivially recovered from `R`. The final components of the PSGS are the CFGs with synchronized rules.

We show an execution of our PAL algorithm below. For two iterations, we show the set of strings `A`, the sets of production rules P_i, and the relation `R` with the synchronized rules. The input set of strings is derived from the previous example of PSGS with an additional spurious string A_4. Dashes are just for visual presentation of the time period associated with each symbol in `A`. Non-terminals are displayed only with their index numbers.

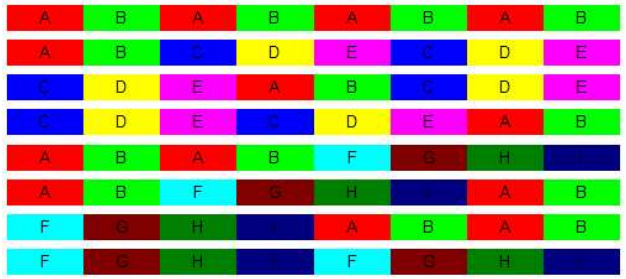
```
A = {(a-5d-5ada-5d-5ada-5d-5ad),
     (-1ada-1d-1ada-1d-1ada-1d),
     (--4---4---4---4---4---4-),
     (adadcabcadbbdbcacdcbbaad),
     (a-6d-6ada-6d-6ada-6d-6ad)},
P1 = {5 -> bc},
P2 = {1 -> bc},
P3 = {2 -> cd, 3 -> 2a, 4 -> 3b},
P4 = {},
P5 = {6 -> bc},
R = {(2, 1), (3, 1), (4, 1), (5, 1), (5,
2), (5, 3), (5, 4), (6, 1), (6, 2), (6, 3),
(6, 4), (6, 5)}.
```

```
A = {(a-5d-5ada-5d-5ada-5d-5ad),
     (-1ada-1d-1ada-1d-1ada-1d),
     (----7-------7-------7---),
     (adadcabcadbbdbcacdcbbaad),
     (a-6d-6ada-6d-6ada-6d-6ad)},
P1 = {5 -> bc},
P2 = {1 -> bc},
P3 = {2 -> cd, 3 -> 2a, 4 -> 3b, 7 -> 44},
P4 = {},
P5 = {6 -> bc},
R = {(2, 1), (3, 1), (4, 1), (5, 1), (5,
2), (5, 3), (5, 4), (6, 1), (6, 2), (6, 3),
(6, 4), (6, 5)}.
```

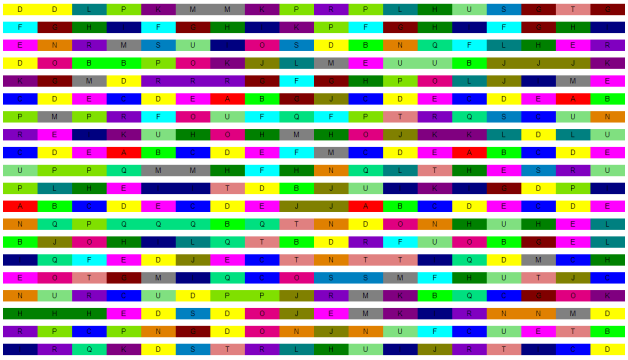
Initially, synchronization is difficult to be detected in practice for low-level non-terminals (closer to the leaves of the grammar tree forest) because they have a high frequency and some atom occurrences are spurious. However, high-level non-terminals are more robust and synchronization is reliably detected for them. In order to overcome this problem, the algorithm could be adapted with a re-check for synchronization. When synchronization is issued for a pair of non-terminals *A* and *B*, their descendents in the respective grammar trees are re-checked for synchronized rules. This time, considering only instances of their descendent non-terminals which are concurrent with *A* and *B*, respectively.

Besides formally specifying the relations between CFGs, the synchronized rules are effective in identifying the maximum level of generalization for an action as demonstrated with the non-terminal `7` above. Further, the set of strings related by synchronized rules corresponds to the actual grammar components. The basic idea is to eliminate non-terminals with no associated synchronization and the resulting grammars are the true components of the learned PSGS. Note that the grammar associated with string A_4 above will end up with only three non-synchronized rules (`P4 = {8 -> ad, 28 -> ca, 29 -> bb}`), which correctly identifies it

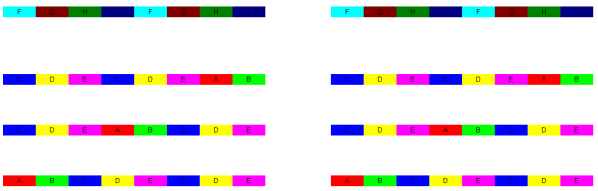
as the spurious string not belonging to the grammar system inferred.



(a) Pattern templates



(b) Synthetic actiongram



(c) Ground truth

Fig. 4: Evaluation with synthetic data.

We evaluated our PAL algorithm with synthetic data and real human motion data. A synthetic actiongram was created with 20 synchronous strings, each one containing 100 uniform segments. Each segment is associated with a symbol extracted from an alphabet of 20 characters. Four synchronous strings in the actiongram are created according to a pattern chosen among one of eight different templates (see Fig. 4a). These templates are repeated 10 times along the patterned string (separated by two random characters) to represent a consistent movement performed several times. Different templates are applied to the four patterned strings synchronously. The remaining strings are generated with random symbols from the alphabet in order to simulate spurious movement (see Fig. 4b).

The ground truth for our problem is available in a synthetic actiongram (see Fig. 4c). We compare the output of our algorithm with this ground truth in order to define an evaluation criterion. If the output matches the ground truth, i.e. all four pattern strings are identified and the corresponding templates are extracted, we claim the algorithm was successful.

For a more realistic evaluation, we inserted noise in the synthetic data. The four patterned strings have a number of symbols replaced by noisy random characters in the alphabet. We tested our algorithm 100 times for an increasing level of noise and computed the overall success rate for each noise level (see Fig. 5). The algorithm achieves 100% success rate up to 7% of noise inserted in the patterned strings. The algorithm is robust even at 10% of noise level when the success rate was 96%.

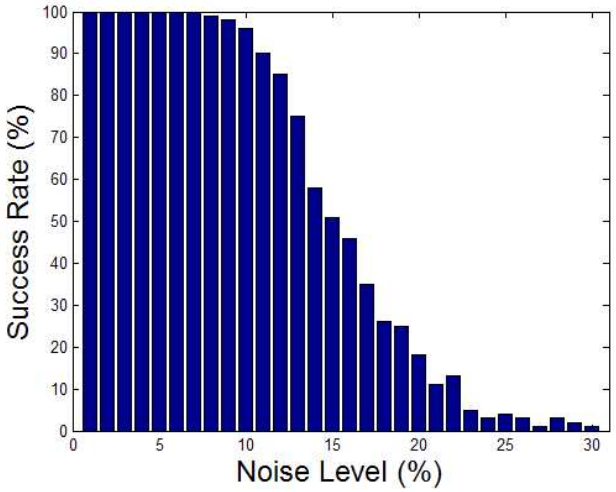


Fig. 5: Evaluation with increasing noise levels.

4. Human Activity Language

Besides sensory-motor primitives, we suggest five kinetological properties (compactness, view-invariance, reproducibility, selectivity, and reconstructivity) to evaluate them. In [9], we discuss these principles in detail and evaluate our segmentation method and primitives according to them.

4.1. Kinetology

The compactness principle is related to describing a human activity with the least possible number of atoms. Compactness is achieved through segmentation which reduces the number of parameters in the representation. Our segmentation approach was implemented as a compression method for motion data and resulted in files with about 3.698% of the original size in our motion database.

An action representation should be based on primitives robust to variations of the image formation process. View-invariance regards the effect of projecting a 3D representation of human movement into a 2D

representation according to a vision system. A view-invariant representation provides the same 2D projected description of an intrinsically 3D action captured from different viewpoints.

The view-invariance evaluation requires a 2D projected version of the initial representative function according to varying viewpoints. A circular surrounding configuration of viewpoints is used. A view-invariance graph shows for each time instant (horizontal axis) and for each viewpoint in the configuration of viewpoints (vertical axis), the state associated with the movement (see Fig. 6).

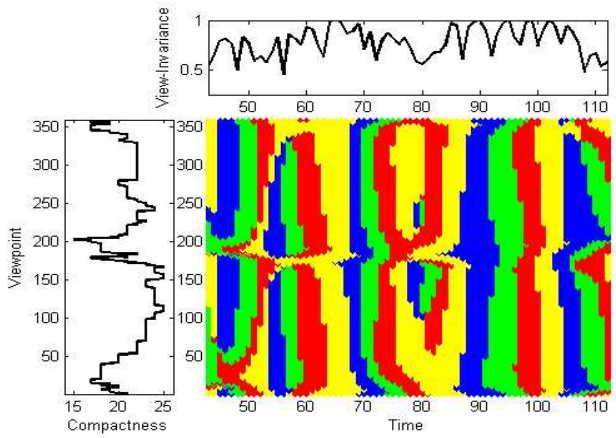


Fig. 6: View-invariance of the left knee flexion/extension angle.

The view-invariance is affected by some uncertainty at the borders of the segments (temporal dimension) and at degenerate viewpoints (viewpoint dimension). However, the states of movement segments are stable for most of the time according to viewpoint variability.

Reproducibility requires an action to have the same description even when a different performance of this action is considered. Intra-personal invariance deals with the same subject performing the same action repeated times. Inter-personal invariance concerns different subjects executing the same action several times. A kinetological system is reproducible when the same symbolic representation is associated with the same action performed at different occasions or by different subjects.

In order to evaluate the reproducibility of our kinetological system, we used a human gait database with 16 subjects covering males and females at several ages. A reproducibility measure is computed for each joint angle. The reproducibility measure of a joint angle is the fraction of the most representative symbolic description among all descriptions for the database. The reproducibility is very high for the joint angles which play a primary role in the walking action (see Fig. 7). The identification of the intrinsic variables of an action is a byproduct of the reproducibility requirement of a kinetological system.

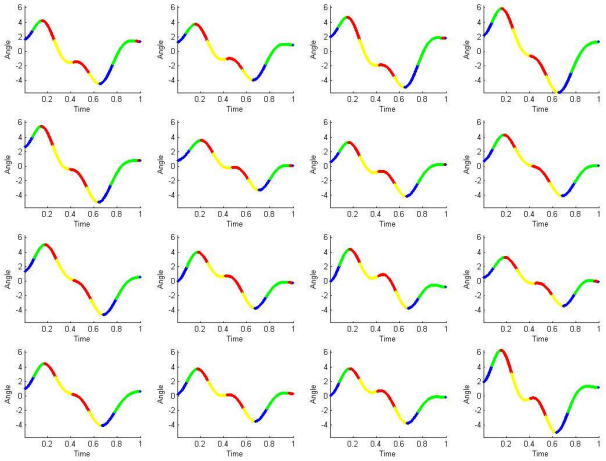
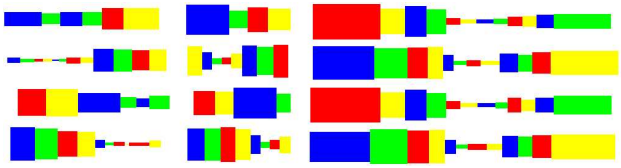


Fig. 7: Reproducibility of the pelvic obliquity during gait.

The selectivity principle concerns the ability to discern between distinct actions. In terms of representation, this principle requires a different structure to represent different actions. We compare the compact representation of several different actions and verify whether their structures are dissimilar.



(a) Walk (b) Run (c) Jump

Fig. 8: Selectivity: different representations for three distinct actions.

The selectivity property is demonstrated in our representation using a set of actions performed by the same individual. Four joint angles are considered: left and right hip flexion-extension, left and right knee flexion-extension (see Fig. 8). The different actions are clearly represented by different structures.

Reconstructivity is associated with the ability to reconstruct the original movement signal up to an approximation factor from a compact representation. We propose a reconstruction method based on a novel interpolation algorithm which considers the kinetological structure.

We consider one segment at a time and concentrate on the state transitions between consecutive segments. Based on a transition, we determine constraints about the derivatives at border points of the segment. Each possible sequence of three segments corresponds to two equations associated with first and second derivatives at border points.

A simple model for the joint angle function during a segment is a polynomial. The least degree polynomial satisfying all the constraints is a fourth degree polynomial. This way, the reconstruction process needs to find five

parameters defining this polynomial. The polynomial is partially determined with the two associated equations for the particular sequence of kinetemes and two more equations using the joint angle values at the two border points. These values are obtained from the time length and the angular displacement of each segment. The last free variable can be determined using some criteria such as jerk minimization.

We implemented this reconstruction scheme as a decompression method for motion data (see Fig. 9). The average error for all joints in our motion database was about 0.823°. Once a reconstruction scheme is implemented, the generation of movement from a symbolic representation is feasible. Therefore, the symbolic grammar systems inferred for human actions may be used to generate movement.

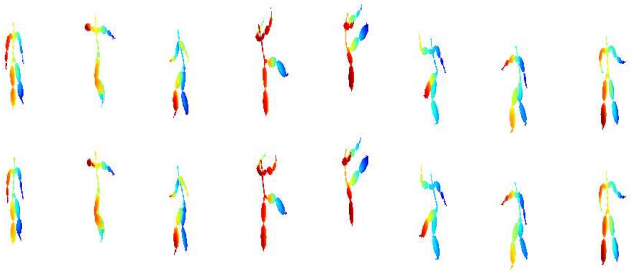


Fig. 9: Reconstructivity: original and decompressed sequences of same activity.

4.2. Action Morphology Inference

The morphology of a human action consists of the subset of true actuators, their corresponding motion patterns and CFG components, and the synchronized rules among them. Ground truth information for this kind of human movement representation is not available. The identification of which joint actuators are involved in a single action such as walking forward is a challenging problem. The answer for this question would involve a research project in Kinesiology. This way, without access to ground truth data for real human motion modeling, we validated our approach with a large scale motion capture database.

The whole grammatical inference process is data driven. In order to find movement patterns, we infer a PSGS from a symbolic representation of real human movement data: an actiongram. The motion data consists of the repeated execution of the same action. However, the action is not required to be a repetitive activity. This process was performed in several actions separately. We tested our algorithm in a motion capture database with about 200 different general actions. The subset of induced grammar components is associated with joint angles concerned intrinsically with the action. The resulting grammars represent the morphological structure of the action being induced.

Given an actiongram representing the same action (e.g. walk, throw, push) performed repeated times. We apply our PAL algorithm to this symbolic representation in order to infer a PSGS modeling this action. Using the synchronized rules, we prune spurious production rules in the component grammars. Consequently, the remaining rules serve to identify the subset of true components related to the action. The resulting component grammars correspond to the actuators coordinated for the achievement of a common purpose embedded in the action. Overgeneralized rules are also discarded due to the lack of synchronization. Therefore, the remaining highest-level in each grammar component delimits the motion pattern associated with the action.

Fig. 10: Actuator sets extracted from about 200 actions in our database.

The joint actuators active during the execution of a human activity are represented by a binary string (see Fig. 10). From the morphemes of our motion database, we have extracted a set of about 200 binary strings representing these actuator sets in the most basic level for each action.

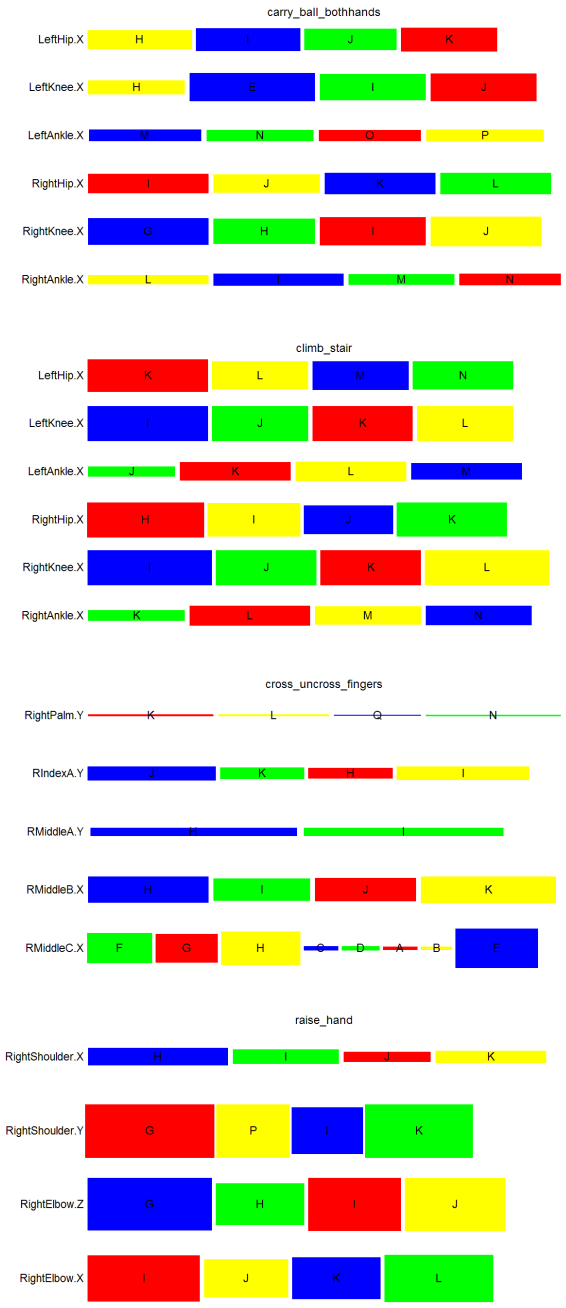
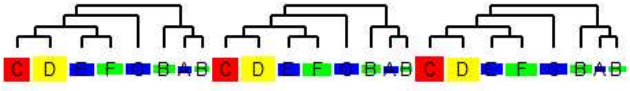


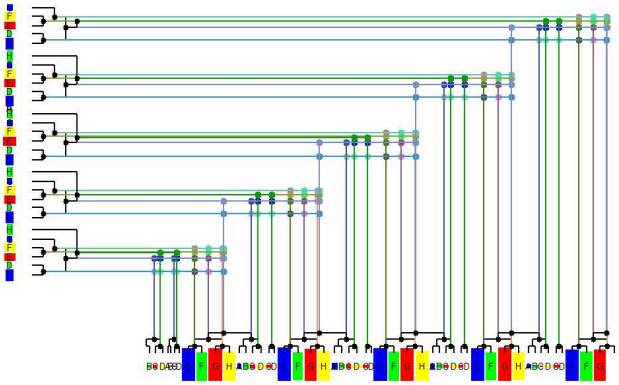
Fig. 11: Sample morpheme examples learned for human actions.

Parallel learning selects a subset of the actiongram which projects the whole action only into the intrinsic joint angles and motion patterns of the action. We successfully identified the morphemes in each action of our motion database, i.e., joints participating in each action, the motion patterns (kinetemes), and their synchronization with movement in other joints. In Figure 11, we show a sample of morphemes inferred from real motion data with our technique.

Additional structure in our modeling of human action is a CFG as a component grammar for each true actuator. This grammar corresponds to a forest of binary trees. The starting symbol of a component grammar is the non-terminal associated with the root nodes in the forest (see Fig. 12a). Coordination between different actuators is represented by synchronized rules (see Fig. 12b).



(a) A CFG component



(b) Two CFGs (hip and knee) related by synchronized rules

Fig. 12: A PSGS structure for human movement.

Given the morphology of each action in our database, we may infer additional structure on the morphemes of movement. Further learning of the most frequent sets of joints that are active in all actions and the corresponding initial poses will lead to higher-level organization. In this sense, motion patterns of different action for a particular joint actuator may have a common structure. Some motion patterns share the same kineteme (depicted as black segments in Figure 13). This way, the morphological grammars become even more compact with just a few kinetemes required to represent all motions.

Our framework was able to infer movement patterns that closely model the original movement. The patterns provide high-level and explicit information about the meaning of each human activity. Therefore, our approach was successful in both representational and learning aspects, serving as tool to parse movement, learn patterns, and to generate actions.

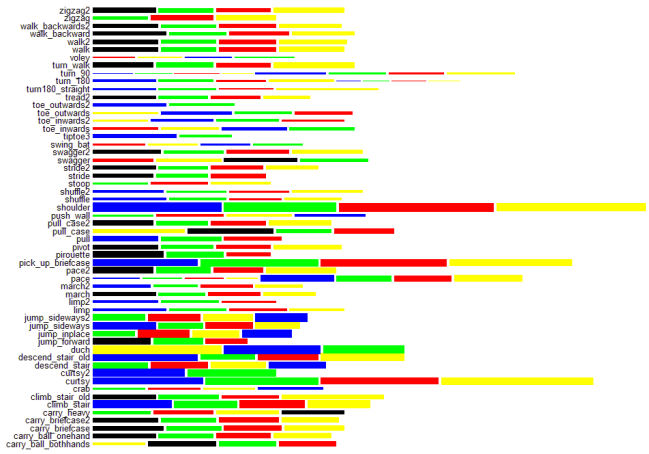


Fig. 13: Motion patterns for the left knee in our database.

5. Conclusion

We presented a human movement representation considering the variability in the set of active joints for different activities. Our representation explicitly contains the set of joints (degrees of freedom) actually responsible for achieving the goal aimed by the activity, the synchronization rules modeling coordination among these actuators, and the motion performed by each participating actuator.

We discussed the morphological part of our linguistic framework for the modeling and learning of human activity representations. In this part, we associate each action with a formal grammar system. The formal model proposed is an adapted Parallel Communicating Grammar System (PCGS), named Parallel Synchronous Grammar System (PSGS), which is induced with a new PArallel Learning (PAL) algorithm.

Our algorithm is used in the learning of human activity and induces, as components, the actuators responsible for the achievement of the purpose associated with the activity. Instead of a sequence of segmented sentences, the input is a single string for each actuator in the body. This string represents the language to be inferred and the algorithm infers a CFG grammar for each component in the grammar system.

The heuristic inference of a parallel grammar system resolves overgeneralization issues usually found in sequential learning. Our algorithm infers a simplified Parallel Communicating Grammar System without any structural information about the components or component languages.

Towards the discovery of a sensory-motor Human Activity Language (HAL), we presented the steps required for the construction of a praxicon. A praxicon is the kinematic analogous of a lexicon in spoken language. We intend to learn a large praxicon through the inference of the grammar systems corresponding to a large set of actions. The learned templates of human action will allow the mining of strategies of movement. This will lead to the syntax of human activity, another part of our linguistic framework, and will have implications in the parsing of human action.

Another important issue concerning the non-arbitrary mapping of motion data to concrete concepts associated with human action is the grounding of symbolic reasoning systems. A logic-based conceptual system is grounded in sensory-motor information through this mapping. Therefore, our linguistic framework is another way to attach meaning to a conceptual reasoning system.

6. References

[1] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data", in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 375-381, 2003.

[2] M. Brand and A. Hertzmann, "Style machines", in *Proc. of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 183-192, 2000.

[3] E. Csuhaj-Varjú and J. Dassow, "On Cooperating/Distributed Grammar Systems", *Journal of Information Processing and Cybernetics*, vol. 26, no. 1-2, pp. 49-63, 1990.

[4] H. Fernau, "PC Grammar Systems with Terminal Transmission", *Acta Informatica*, vol. 37, no. 7, pp. 511-540, 2001.

[5] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti, "Action Recognition in the Premotor Cortex", *Brain*, vol. 119, no. 2, pp. 593-609, 1996.

[6] A. Glenberg and M. Kaschak, "Grounding Language in Action", *Psychonomic Bulletin & Review*, vol. 9, no. 3, pp. 558-565, 2002.

[7] E. Gold, "Language Identification in the Limit", *Information and Control*, vol. 10, no. 5, pp. 447-474, 1967.

[8] G. Guerra-Filho and Y. Aloimonos, "Towards a sensorimotor WordNet^{SM}: Closing the semantic gap", in *Proc. of the International WordNet Conference (GWC)*, pp. , 2006.

[9] G. Guerra-Filho and Y. Aloimonos, "Understanding visuo-motor primitives for motion synthesis and analysis", in *Proc. of Computer Animation and Social Agents (CASA)*, 2006.

[10] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[11] G. Hoyle, *Muscles and their Neural Control*, John Wiley, New York, 1983.

[12] Y. Ivanov and A. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852-872, 2000.

[13] O. Jenkins and M. Matarić, "Automated derivation of behavior vocabularies for autonomous humanoid motion", in *Proc. of the International Conference on Autonomous Agents*, pp. 225-232, 2003.

[14] R. Meersman and G. Rozenberg, "Cooperating Grammar Systems", *Lecture Notes in Computer Science*, vol. 64, pp. 364-374, 1978.

[15] F. Mörchen, A. Ultsch, and O. Hoos, "Extracting Interpretable Muscle Activation Patterns with Time Series Knowledge Mining", *International Journal of Knowledge-Base and Intelligent Engineering Systems*, vol. 9, no. 3, pp. 197-208, 2005.
[16] C. Nevill-Manning and I. Witten, "Identifying Hierarchical Structure in Sequences: A Linear-Time Algorithm", *Journal of Artific. Intelligence Research*, vol. 7, pp. 67-82, 1997.
[17] N. Nishitani, M. Schurmann, K. Amunts, and R. Hari, "Broca's Region: From Action to Language", *Physiology*, vol. 20, pp. 60-69, 2005.
[18] R. Parekh and V. Honavar, "Grammar inference, automata induction, and language acquisition", in R. Dale, H. Moisl, and H. Somers, editors, *The Handbook of Natural Language Processing*, Marcel Dekker Inc., pp. 727-764, 2000.
[19] G. Păun, "On the Synchronization in Parallel Communicating Grammar Systems", *Acta Informatica*, vol. 30, no. 4, pp. 351-367, 1993.
[20] G. Păun and L. Sântean, "Parallel Communicating Grammar Systems: The Regular Case", *Annals of the University of Bucharest*, *Mathematics-Informatics Series*, vol. 38, no. 2, pp. 55-63, 1989.
[21] H. Sidenbladh, M. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking", in *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 784-800, 2002.
[22] P. Sosík and L. Štýbnar, "Grammatical Inference of Colonies", *Lecture Notes in Computer Science*, vol. 1218, pp. 236-246, 1997.
[23] J. Stuart and E. Bradley, "Learning the grammar of dance", in *Proc. of the International Conference on Machine Learning (ICML)*, pp. 547-555, 1998.
[24] J. Yang, Y. Xu, and C. Chen, "Human Action Learning via Hidden Markov Model", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, no. 1, pp. 34-44, 1997.
[25] T.-S. Wang, H.-Y. Shum, Y.-Q. Xu, and N.-N. Zheng, "Unsupervised analysis of human gestures", in *Proc. of the IEEE Pacific Rim Conference on Multimedia*, pp. 174-181, 2001.
[26] J. Wolff, "Learning syntax and meanings through optimization and distributional analysis", in Y. Levy, I. Schlesinger, and M. Braine, eds, *Categories and Processes in Language Acquisition*, Lawrence Erlbaum Assoc., Inc., Hillsdale, NJ, pp. 179-215, 1988.