ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/51818102

The minimalist grammar of action

Article *in* Philosophical Transactions B · January 2012

DOI: 10.1098/rstb.2011.0123 · Source: PubMed

CITATIONS
130

READS
846

2 authors:

Katerina Pastra
Athena-Research and Innovation Center in Information, Communication and Kno…
56 PUBLICATIONS **501** CITATIONS
SEE PROFILE

Yiannis Aloimonos
University of Maryland, College Park
453 PUBLICATIONS **11,995** CITATIONS
SEE PROFILE

All content following this page was uploaded by Yiannis Aloimonos on 26 May 2014.

The user has requested enhancement of the downloaded file.

The Minimalist Grammar of Action

Katerina Pastra[1] *and Yiannis Aloimonos*[2]

[1]*Cognitive Systems Research Institute, 7 M. Prantouna Str., 11525, Athens, Greece, kpastra@csri.gr*

[2]*Computer Vision Lab, University of Maryland, College Park, MD20742, USA, yiannis@cs.umd.edu*

Abstract

Language and action have been found to share a common neural basis and in particular a common "syntax", an analogous hierarchical and compositional organization. While language structure analysis has led to the formulation of different grammatical formalisms and associated discriminative or generative computational models, the structure of action is still elusive and so are the related computational models. However, structuring action has important implications on action learning and generalisation, in both human cognition research and computation. In this paper, we present a biologically inspired generative grammar of action, which employs the structure-building operations and principles of Chomsky's Minimalist Programme as a reference model. In this grammar, action terminals combine hierarchically into temporal sequences of actions of increasing complexity; the actions are bound with the involved tools and affected objects and are governed by certain goals. We show, how the *tool-role and the affected-object role* of an entity within an action drives the derivation of the action syntax in this grammar and controls recursion, merge and move, the latter being mechanisms that manifest themselves not only in human language, but in human action too.

Key Index Words

Generative grammar of action, tool use, action syntax, action decomposition, temporal sequence, minimalist grammar

Introduction

The repertoire of human actions is infinite, starting from the simplest intentional body movements such as *stretching a leg* to creative *dancing routines,* to interaction with tools and objects such as *grasping a knife,* to even more complex series of actions that formulate events, such as *preparing a salad* or *cleaning the house.* Uncovering the structure of action has been a quest in many disciplines, including cognitive science and artificial intelligence. How could one generate or parse actions of any complexity avoiding at the same time overgeneralization? The question is similar to an analogous problem in language analysis: How could one generate or parse all and only the grammatical sentences of a language?

The quest for the structural principles of visual and motoric action goes back at least to the early fifties and suggestions by the psychologist Karl Lashley that syntax may apply not only to language but also to other forms of behaviour, such as goal directed action [1]. From another perspective, the archaeologist Andrė Leroi-Gourhan argued that bipedality led to technology and technology (tool making and use) reflects a capability (for derivation of structures) that may link human action and language [2]. Since then, corroborating experimental evidence on the relation between action and language and the hierarchical structure of action, in particular, abound; for example, two year old children have been found to be able not only to parse hierarchically organized actions [3], but also to copy and reproduce such actions [4]. Complex action structure (analysed as means-end parse trees) has also been found to be represented abstractly, i.e. independently of the actual semantics of the actions [5]. More strikingly, neurobiological evidence on the nature of neural circuits in the traditionally related to language-production area of the human brain (i.e. Broca's area) provides a growing number of suggestions regarding the characteristics of an action grammar, such as the role of body parts/effectors, of tools and object type, and the role of the notion of 'goal' in human action representation [6, 7].

However, specifying an action grammar that will generate thousands of actions is still elusive. There are only very few attempts for developing an action grammar in computational research [8, 9, 10] and these are recognitive rather than generative approaches. There is a need for developing a generative grammar of action that will have both computational expressivity and simplicity, and a biological basis; the former will allow for employing the grammar in artificial intelligence applications, while the latter may prove to be the key for action learning and generalisation.

In this paper, we employ a formal language analysis framework as a reference model for presenting a generative grammar of action. In particular, we employ the generative grammar framework, for crossing over human language, to human action. Though there is a variety of grammars for describing the structure of language, we choose the Chomskyan approach and its latest evolution into the Minimalist Program [11], because it is the culmination of an attempt to describe and explain language syntax in terms of more general principles and operations that are not tightly tied to the idiosyncrasies of the human language system, but instead may have counterparts in other biological systems [12]. This perspective allows one to look for universals not only within the structures of different human languages, but also across natural language to non-symbolic sensorimotor spaces, such as human action.

We present the characteristics and components of this grammar of action, many aspects of which are corroborated by neurobiological findings. We argue that the notion of *'tool-use'* drives action syntax derivation and through examples we present how this takes place when employing the suggested action grammar.

Research on the structure of action

The structure of visual and motoric action has been explored by a number of disciplines, including neuroscience, psychology, computer vision and robotics. In this section, we make a concise presentation of neurobiological and computational research related to the existence and implementation of a grammar of action.

Neurobiological approaches on action grammar

Recent years have seen an increasing body of experimental evidence suggesting that Broca's area, the human brain area traditionally linked to language production, is involved in representing complex hierarchical structures regardless of modality, such as those involved in action execution and observation [13]. In other words, Broca's area has been suggested as the neural locus of an action grammar [14], an area where goals are represented and hierarchical motor chains are

planned [7]. The findings indicate a common syntactic-like structure between language and action which has led to speculations that "this capacity evolved from motor and premotor functions associated with action execution and understanding such as those characterizing the mirror neurons" [13].

At a behaviour level, action syntax has been shown to comprise of simpler elements (motor primitives) that are connected to each other either serially or in parallel (i.e. simultaneously) [15, 16], [17], [18, 19]. Researchers have concentrated on the analysis of many different actions, such as reaching and grasping, gait and balance, posture and locomotion. Reaching movements appear to be coded in terms of direction and extent, and appear to be composed of discrete sub-movements, all with a similar stereotypical, serially concatenated shape and overlapping in time [20, 21, 22]. Human and monkey grasping and object manipulation has been studied extensively and has been described as consisting of sub-actions executed as a unified coordinated complex act (cf. for example [23]). Parallel syntax, on the other hand, involves the simultaneous activation of several muscles that produce a torque about a joint or a force in a particular direction. Electromyogram (EMG) recordings from frog hind limb muscles have been analyzed to test whether natural behavior shows synergies among groups of muscle activities for an entire set of natural behaviors [18], [24], [25] [26]. Similar attempts have been made to find muscle synergies during human posture and locomotion [27, 28].

In some approaches, motor primitives basically amount to motor schemas or control modules that may be specific to a task; for example, in the 'motor ideas/schemas' approach, coordinated control programs regulate co-activation of perceptual and motor schemas and the passing of action parameters from one to another to determine hand-environment interaction [29, 30]. Within this approach, perceptual schemas represent objects which are involved in an action, while motor schemas represent the actual motor program to be executed.

Sequential or parallel, the combination of action primitives or schemas into complex actions has been explored, but has not led yet to a grammar that will allow one to generate thousands of actions, incorporating the ever growing body of related biological evidence.

Computational grammars of action

At a computational level, there is really not much previous work on the subject, i.e. on a computational **motoric grammar** for action. A system that comes closest in spirit to a grammar for action was developed in [31] more than fifteen years ago for handling eye movements. By turning the eye movement data into a string of symbols, they developed a finite automaton (the equivalent of a regular grammar) for representing the data. However, some researchers have come close to the idea of motoric primitives of action and primitives are, indeed, the first step to a grammar. A number of data transformations have been employed to derive a limited number of motor primitives that are then combined through a well-defined set of rules to form more complex actions (see for example the movements of [32] or the modules of [33]). Primitives in these cases may be kinematic, dynamic or kinematodynamic [16, 18, 34, 35] and are extracted using statistical techniques like PCA (principal component analysis) or HMM (hidden markov models), and others.

In a recognitive (rather than generative) approach to action analysis, decomposition of action sequences into primitives has taken many forms. Finger movements and forces have been decomposed into basic synergies based either on the idea of uncontrolled manifold or on inverse dynamics computations [36, 35]. Hand gestures also consist of primitives or more complicated sequences that can be decomposed into a series of more elementary units of activity [37]. In [38], primitives were extracted by k-means clustering the projection of high-dimensional segment vectors onto a reduced subspace, while in [39] the local minimum in total body force was used to detect segment boundaries. In [40], similarities of motion segments were measured according to a dynamic programming distance and clustered with a nearest-neighbor algorithm. In [41] gestures were segmented with the local minima of velocity and local maxima of change in direction. These segments were hierarchically clustered into classes using Hidden Markov Models to compute a metric. Grammar induction techniques were applied to both motion capture data and images (silhouettes) to produce a human activity language [10, 42] thus formalizing and unifying several prior approaches.

The development of a generative grammar for action, i.e. one that can be used both for visual action analysis and generation of goal-directed behaviour is of primary importance for both computer vision and robotic applications. Clearly, such grammar is not only missing in state of the art computational approaches to action analysis, but is also elusive in a formal analysis (theoretical) level. This is the contribution of our paper, the development of a generative grammar of action and in particular, of a grammar with computational applicability and with biological

bases, the latter being for us a prerequisite for scalability and generalisation of a computational approach.

How is action structured?

In order to answer this question, we employ a formal analysis framework that has been developed for language. It is the latest formulation of the Chomskyan tradition of generative grammars, the minimalist program [11]. The minimalist program and the generative grammar paradigm in general have, indeed, many details and intricacies for dealing with a number of phenomena in language. There are many ways of implementing the theory and representing information in the parse trees, with versions of the theory before the minimalist program being implemented and elaborated more extensively. In this paper, we do not wish to go into the details of the representation and the theory, or to follow strictly one or another approach in parse tree representation. Our aim is to present the basic framework, so that it becomes obvious how we employ it as a reference model for developing a grammar of action. Therefore, in this section, we will first introduce this formal analysis approach and then we will present our use of the principles and syntactic operations described in the framework to formulate a generative grammar of action.

The Chomskyan tradition of generative grammars

Generative grammars have been used extensively for the analysis of the structure of human language. Simply put, a generative grammar comprises a set of elements and a set of production (rewrite) rules that correctly predict which combinations of elements form grammatical sentences. A particular type of generative grammars are the phrase structure grammars or else *context-free grammars*, which have recursive rules, i.e. they allow for nesting of elements in same type elements, accommodating thus for embedded structures. These grammars comprise a set of terminals (e.g. lexical categories such as noun, verb, adjective), a set of non-terminals (i.e. the phrases, such as noun phrase, verb phrase etc.) and a set of production rules of the form $X \rightarrow y$, where X is a single non-terminal symbol, and y is a string of zero or more terminals and/or non-terminals. The context of X within a structure does not affect the use of the corresponding rule

(hence context-free). In applying the grammar for the analysis of a certain structure, a parse tree is produced, in which non-terminal symbols are the nodes, terminal symbols are the leaves, and each node expands (through successive application of the production rules) into the next level of the tree [43].

Though highly expressive, this type of grammar cannot account for natural language phenomena such as agreement (e.g. case, number, gender agreement) and reference (e.g. anaphora, relative clauses). These are cases of either 'discontinuous elements' or long-distance dependencies between constituents of a sentence [44, 45]. The Chomskyan tradition of *generative grammar* deals with such phenomena through the use of a number of processes (transformations) on the output of context-free grammars [46]. The latest evolution of the Chomskyan grammar tradition is the Minimalist Program (MP) [11], a framework that reduces transformation grammar to a simple, powerful computational mechanism imbued with the principle of economy/minimalism in both derivation and representation of syntactic structures; this minimalism advocates that: (a) *minimal derivation processes* run for producing the syntactic structure (only those transformations needed to fully interpret the constituents of the structure), and (b) *minimal representations of syntactic structures are produced* (only what is needed to satisfy grammaticality).

The language that has a generative grammar consists of:

(a) a finite set of terminals T, i.e. leaf nodes in a parse tree, or else minimal projections, the actual lexical units that make up a sentence; in the minimalist program, these are characterised through a number of morphosyntactic features F, such as their part of speech, case, type of complement etc.

(b) a finite set of non-terminals NT, i.e. phrase types, syntactic categories of the terminals such as noun phrases, verb phrases etc., such that $T \cap NT = \emptyset$

(c) a finite set of production rules R, i.e. rewrite rules that are applied to terminals and non-terminals recursively (i.e. a rule rewrites as a previous rule or as itself) producing an infinite number of grammatical structures (cf. *Table 1*).

General case	**Instantiation for Sentences**	**Instantiation for Noun Phrases**	**Instantiation for Verb Phrases**
X'' → (y), X'	S'' → (y), S'	N'' → (y), N'	V'' → (y), V'
X' → y, X'	S' → y, S'	N' → y, N'	V' → y, V'

X'→ X', y	S' → S', y	N' → N', y	V' → V', y
X' → X, (y)	S' → S, (y)	N' → N, (y)	V' → V, (y)

Table 1: Generative Grammar Rules

The generative grammar rules in *Table 1* work as follows; X is the minimal projection of a language unit (the actual word) none of the features of which have been checked (has been attributed a value). X' is an intermediate level projection in which some of the features of the unit have been checked, and X'' is the maximal projection of the unit, in which all its features have been checked. Y is a specifier when it precedes X, i.e. a terminal or non-terminal that modifies the meaning of X, and a complement when it follows X, i.e. a terminal or non-terminal that complements the meaning of X. Parenthesis denotes that its presence is optional. The table shows that such grammars can be used for the derivation of complex language structures involving sentences within sentences (second column) which may be analysed down to the level of noun and verb phrases forming a sentence. The structure of noun and verb phrases themselves is also analysed in terms of maximal projections (third and fourth coloumn respectively).

In the MP framework, the derivation of a syntactic structure starts *bottom-up*; a *Merge* function checks the features of a terminal (lexical unit) and for those features with un-attributed values (i.e. variables), it initiates a *Search* for another unit whose feature-values can be unified with the variables. This merging creates binary structures and is applied recursively until all features are 'interpreted' (have a value) [47]. For example, in a simple request such as 'grasp the knife', the verb 'grasp' has an object-complement feature OC with category type 'nominal', case 'accusative' and semantic type 'graspable object'; merging initiates a search for a lexical item with these features, i.e. [+nominal, +accusative, +graspable] in order to fill in the object complement variable OC. The determiner 'the' initiates a further search for satisfying its own features, which leads to the creation of the noun phrase 'the knife'. This noun phrase can now satisfy the 'grasp' search for an element or structure that interprets its own feature variables, so a further merging takes place. Every merged set of elements (phrase) has a label (the head of the phrase) that determines the properties of the phrase, e.g. {the {the, knife}}, {grasp{grasp,the knife}}. These properties allow certain projections and eliminate alternatives.

One form of merging is the *Move*-operator. It is 'merging' of one morpho-syntactic element with itself (internal merging). For internal merging to take place, a probe-goal relation must hold between at least one feature of an element and a corresponding feature of another element [11].

For example, in the sentence "which knife$_i$ has John grasped $\varnothing_i$?" there is an internal merging between the moved element 'knife' (normally expected after the verb, where the 'null' element with trace 'i' is) and its co-indexed trace (i).

A minimalist grammar of action

In employing a generative grammar for the analysis of the structure of human action, one needs to define the set of terminals, features, non-terminals, and production rules in the sensorimotor domain. So, which are these terminals and non-terminals, which are their 'morpho-syntactic' features, and how do they merge creating more and more complex actions?

In what follows, we will present a minimalist grammar of action which consists of action primitives (terminals), action 'phrases' (non-terminals), and three action features that drive merging, namely the *tool-complement* of an action, the *affected-object* complement and the *goal* of the whole action structure.

[Figure 1]

In our analysis, we consider a human action to be a *serial or parallel conjunction of perceptible movements carried out by one or more actors with a certain goal*. We identify three main 'morpho-syntactic' features which characterize human actions and which we employ for defining action terminals and non-terminals. These 'morpho-syntactic features' are actually parameters that affect the execution of actions and distinguish one *action type* from another; they go beyond movement execution features (e.g. direction, velocity etc.) which they modify though, as soon as a movement is embedded within action context. We will go through these parameters, referring in parallel to an example action parse tree shown in Figure 1:

(a) Tool complement (t_c): the effector of a movement, this being a body part, a combination of body parts or the extension of a body part with a graspable object used as a tool.

Actions are always being executed through the use of an effector (body part) or its extension, i.e. an instrument (artifact). Being a body part or artifact, the 'tool' used changes the execution of the action in terms of configuration of the effector, force exerted etc. For example, grasping

something with the hand is different (in its motoric execution and complexity) from grasping it with pliers, which is different from grasping it with tweezers and so on. What we refer to here is a merging in the motoric space, in which any action necessarily/inherently requires a 'tool complement' so, a search for the entity that interprets this variable is initiated for deriving a first binary action structure. See for example, the action-tool binary branches of the tree in Figure 1.

A*ny graspable entity* can be used as a tool in the realization of an action (e.g. use of a book to pound something). In some cases, the particular use of the entity is common (e.g. use of a hammer for pounding), in other cases it may be uncommon, but still possible (e.g. the case of book for pounding). We consider this an essential feature of any human action, which is syntactic, i.e. it is explicitly present in the perception or execution of an action, as an independent constituent.

We have to note that under the notion of a 'tool complement', we include both body parts and artifacts/instruments, suggesting essentially that our body part effectors are tools (means for performing actions) in the same way as other objects-artifacts may be used as 'means' for achieving a task. This is corroborated by neurobiological experiments which indicate that tools are indeed perceived as extensions of one's own body part [48, 49], so there is an intricate relation between body parts and other objects through the attribution of a 'tool role' to them. Recent experimental findings have also shown that in visual information processing, humans differentiate graspable objects consistently faster than non-graspable ones, and among graspable ones, prototypical tools (e.g. hammer) are differentiated faster than natural kinds (e.g. carrot) [50]; more importantly, this differentiation was found to be mediated by the activation of motor areas (cf. also [6], on the visuomotor neurons). This evidence suggests that the 'toolness' of an object may be an important differentiation parameter in an action grammar.

(b) Object complement (o_c): any object affected by a tool-use action.

This is another syntactic feature of action; it is the object of interaction, to which the effects of an action are transferred. It may be any entity. This feature actually calls for a further merge operation, between the action-tool structure and the affected object. It results in {action-tool, object} binary structures such as the ones shown in Figure 1 between branches of the tree related through the action-object relation, e.g. {grasp with hand1, knife}. The object that is affected by an action differentiates the action itself; for example, *grasping a pencil with the hand* is different from *grasping a glass with the hand*, not only because the goal may be different but also because

of the characteristics of the object grasped. Biological evidence of strict congruence of action type with object type (e.g. discharge of 'precision grip neurons' when small objects are observed in canonical visuomotor neurons in area F5) [6] suggests that object-complements are indeed differentiation parameters for actions.

(c) Goal (g): the final purpose of an action sequence of any length or complexity.

Another important feature of actions that modifies their execution is their goal. This is a morphological feature, i.e. it is not an explicit, independent syntactic constituent of the action phrase, but instead an 'inflectional' feature, a parameter that modifies the execution of the constituents of an action in terms of effector configuration and spatial interaction with tool and object complements. The execution of a movement is modified according to the action sequence in which it is embedded, i.e. according to the final goal of the action. An analogy to the phenomenon of agreement in sentences can be drawn here: person and number agreement of words, for example, modify the words of a sentence themselves; for a grammatical sentence, all words must be in agreement. Similarly, in action, all sub-actions must agree in terms of the final goal to be served. Thinking of a word or phrase stripped of any agreement indicators is as artificial as thinking of a purposeless action. This is related to what Luria called 'kinetic melody' [51], i.e. the fluidity of motor acts as they follow one another.

Going back to the example in Figure 1, in the action parse tree, the goal feature is attached to a null-constituent and dominates the whole action sequence and its sub-actions. In producing the action tree bottom up (i.e. as the action evolves in time), the goal feature of the sub-actions remains unspecified; it is only when all other features have been checked and no more actions can be merged into a common complex structure that the goal feature can be checked too, i.e. when the top node of the action tree is reached; the goal is the final action.

This role of the goal feature in our grammar is supported by experimental findings which show that certain neurons discharge during goal directed action only [6], and many of them have categorization, generalisation and specificity characteristics, for example they go beyond effector differences [6, 52]. In [7], it was shown that certain neurons in the monkey brain, go beyond object type differences when the same movements share a goal, and that the intention of an action sequence is reflected already in its first sub-action [7]. The latter was also shown to be the case in the human brain; in [53], a forward activation of motor sequences was shown in typically developing children (as opposed to children with autism). In these experiments, increased activity

of the muscles involved in mouth opening was found before grasping takes place in 'grasping to eat' action sequences (vs. grasping to displace ones), during both action observation and action execution; such activation denotes that the final goal of the action sequence was predicted (and actually 'experienced') from the very first phases of the sequence.

In other words, these findings point to important aspects of the role played by the final goal of an action structure:

(a) The realization of the same movement type with the same tool and object complements changes when the goal of the action changes, e.g. *grasping a pencil in order to displace it* is different from *grasping a pencil in order to write*;

(b) The realization of the same movement type with different tool and/or affected object changes, even if the goal of the action remains the same, e.g. *grasping an apple to displace it* is different from *grasping a cube to displace it* (though in such cases effects of the expected/common goal of an object seem to be present, cf. [7]; and last

(c) The final goal of an action sequence is predicted from the very first sub-action(s) of the sequence; for example, sub-components of the grasping activity such as to '*extend the hand towards the pencil*' involve a configuration of the effector that depends on the final goal. Actually, the corresponding 'grasping neurons' begin to discharge before the object-hand contact [6], while the pre-shaping of the fingers also takes place during the transfer of the hand [30].

We need to note, that this 'goal' feature that governs all constituents of an action structure is the *global goal* (the final goal) of the action structure. One may argue that each sub-action of an action structure may have its own *local/immediate goal* too; for example, in figure 1, extending $hand_1$ towards something has the immediate goal of enclosing in $hand_1$ this object (i.e. grasping it); its immediate goal is the next action that it enables. In Artificial Intelligence, traditional planning techniques for the analysis/execution of a task divide the task into sub-goals, i.e. into steps with their own immediate goals; however, the neurobiological evidence mentioned above points to the fact that the final goal of a complex action is evident even in its very first phases; it affects the motoric execution of the sub-actions and it is evident in the early activation of muscles that are related to final action-constituents of the action sequence. As it will be shown in the next section, our minimalist grammar of action makes no use of sub-goals; instead, it is the final goal of

an action structure that is required for deriving the maximal projection of an action (i.e. A'' in the action tree).

Other 'morpho-syntactic' features of action are modifiers that denote the location/scene an action takes place at, or an object that is used as the location of an action (e.g. 'slice bread on the table'); these do not inherently affect the execution of the action itself, they specify the setup of the action. So, their presence is optional.

The tool and object complements as well as the modifiers are entities; these entities have their own perceptual (e.g. visual or other) grammar the terminals and non-terminals of which could be defined in terms of the action grammar. Elaboration on a perceptual (e.g. visual) grammar of objects is beyond the scope of the current paper; however, some general definitions should be in place:

Entity Terminals: these are the simplest entities (objects) which can be defined as perceptible entities that participate in at least 1 motor program and do not comprise of other entities themselves. They are distinguished from each other through their perceptible features (colour, shape, texture etc.) and the role they play in the motor programs in which they participate (i.e. tool-complement, object-complement, or location-modifier). Body parts and natural kinds are expected to form the set of such entities.

Entity Non-Terminals: perceptible entities that consist of entity terminals in certain spatial configuration. They participate as complements or modifiers in more than one motor program. Scenes are included in this set, and they participate in motor programs as modifiers of actions (i.e. they denote the location in which an action takes place).

Having presented the basic action features, we can now turn to the definition of the '*vocabulary*' of the action grammar:

Action Grammar Terminals: The simplest actions, i.e. perceptible movements carried out by an agent to achieve a goal, which have (one or more) body part tool-complements and no object-complements. They have no action constituents themselves and they may be circular/repetitive. This is the set of all possible human body movements, such as limp, sprint, extend arm, raise hand, stretch leg, open/close hand etc., i.e. the set of intransitive biological actions. Action terminals are further distinguished from each other through their perceptible motor features such as speed,

force and direction. For example, the leaf movement nodes of the action parse tree in Figure 1 comprise of the 'extend' ($hand_1$) terminal and the 'enclose' ($hand_1$) terminal.

Action Grammar Non-Terminals: These are perceptible action phrases, that consist of action terminals (or other non-terminals) in certain temporal configuration; they may have both tool-complements and object complements. They involve interaction with objects beyond one's own body or with other agents, for attaining a particular goal/task, such as grasp_knife, slice_tomato etc. Searching for the value of an action phrase's complement using the values of a subsequent action phrase complement guides the merging process (i.e. it determines the boundaries of the complex action, the top node). The latter implies that we define events, as actions in temporal conjunction (sequential or parallel) that share features (see more details in the following section).

Action Grammar Rules

Having defined the constituent elements of the action grammar, we can now present the production rules. These are presented in *Table 2*:

Rules	
4	A'' →g, A'
3	A' → (m), A'
2	A' → A', (o_c)
1	A' → A, t_c

Table 2: Generative action grammar production rules.

A: action terminal, A': intermediate action structure, A'': maximal action structure, g: goal, m: modifier, t_c: tool-complement, o_c: object-complement, (): optional presence

In *Table 2*, the action grammar production rules express the fact that no matter how simple or complex an action is, it has a compulsory goal specifier and a compulsory tool-complement. The presence of affected object-complements is optional and so is the presence of location-modifiers. One will note that in this grammar, there is no explicit reference to the agent that performs the action, as in the language trees for example, where the actor (subject of the verb) may be denoted explicitly (corresponding NP) or implicitly (through person agreement). This is so, because of the body-part tool complements of the action terminals; an action is inherently performed by an agent and since the terminals of the grammar incorporate necessarily a (human or animal) body-part

complement, the agent information does not need to be explicitly present, as a separate, non-tool complement.

Going back to the action tree in Figure 1, the tree can be derived bottom up, through recursive application of the grammar rules. Traditionally, parsers apply a grammar for the analysis of a certain structure, i.e. their input is the whole structure (e.g. sentence) to be analysed, segmented into tokens (terminals); when more than one sentence is to be parsed (i.e. a paragraph or whole text), automatic segmentation of the text into sentences is also provided in advance. Whereas in the language/symbolic space, the automatic segmentation of a text into sentences (i.e. structures to be considered separately for derivation of syntactic trees) and tokens (terminals) is a straight-forward process due to the discrete nature of the data to be analysed, in the sensorimotor space this is not the case[1]. Segmentation of a continuous stream of visual and/or motoric action into 'sentences' (i.e. groups of sequential or parallel actions that combine into an action tree) is a very challenging task [54]. Tokenisation of such 'action sentences' requires a number of sensorimotor processing technologies to be employed, such as image segmentation, object recognition and action recognition; these technologies face a number of challenges and their output cannot been taken for granted when developing an action parser. On the contrary, a parser that applies the minimalist grammar of action can actually *use the grammar to guide the segmentation of visual and motoric action*:

(a) for 'tokenization': the parser can provide an 'attention-guiding' strategy for selecting those objects in a scene that are related to the actions as tools or affected objects (or even locations), while

(b) for 'sentence segmentation', the parser can use the minimalist action grammar operators and features to decide when merging stops and a new action structure starts.

In what follows, we sketch such dynamic parser, which applies the rules of the minimalist grammar of action bottom up along with related segmentation criteria in order to derive the parse tree(s) of non-previously segmented sensorimotor input:

STEP 1:

- *find the first action A'_1 such that its start time is on/after the start time of the visual/motoric input A_{max}, and it has a body-part tool complement;*

[1] We refer to tokenization in text, on which parsers normally run; in speech, the tokenization difficulties due to the 'continuous' nature of the data processed become more evident.

In other words, get the first body part that is in motion in the input stream A_{max} and keep its motoric characteristics as those pertaining to an action terminal A. This is a merging of an action terminal A with its effector, i.e. an entity that has a semantic type feature the value of which can be unified with the tool-complement feature value of A e.g. A = *'extend'* [$t_{c\text{-}body\_part}$: $Variable_1$] merges with E''= '$hand_1$' [+ body part]. This creates the first binary action structure of the form: A'_1 = {A, E''} e.g. A'_1= *'extend* $hand_1$*'*. Up to this point, *rule 1 of the action grammar* has been applied.

- *Search for an entity that could satisfy an object-complement feature of* A'_1*, i.e. for an object affected by the* A'_1 *action-tool binary structure (and no other action), and perform one more merging, creating the action structure* A'_{1b}*; if no such object complement is present, then a null object-complement is derived.*

This applies rule 2.

- *If an object-complement is present, get its location and create the action structure* A'_{1c} *which comprises* A'_{1b} *and the location as its modifier.*

This applies *rule 3* of the grammar, only in cases when an object complement is present. At this stage the parser does not proceed to applying rule 4, i.e. attributing a goal to the action structure; instead, it checks for what follows in order to decide whether to merge the following actions into a larger action structure:

STEP 2:

- for as long as A_{max} *extends in time beyond the end of the thus far derived structure* A'_{1b} *(i.e. as long as visual/motoric input is fed to the parser), find an action* A'_2 *that follows (or is parallel in time with)* A'_{1b}*, such that* A'_{1b} *and* A'_2 *share the same tool complement, or the tool-complement of one is the same with the object-complement of the other*; if so, relate the two actions through *temporal conjunction* and apply the grammar rules from the start.

For example, in the action parse tree in figure 1, the action '*enclose* $hand_1$*'* (i.e. opening/closing hand for grasping) follows the *'extend* $hand_1$*'* action; the fact that they share the same tool-complement (i.e. $hand_1$) indicates that the two of them together form a more complex action unit. This is used as a criterion for continuing the merging of subsequent actions into the same action tree. *Rule 1 is applied* for the formation of A'_2 and rule 2 is applied too for interpreting all its complement features expanding it into A'_{2b}. So, a further merging takes place between the

'*enclose hand*$_1$' structure and the object that interprets its object-complement feature, the 'knife'. Therefore, the *A'*$_{2b}$ action structure derived is '*enclose* with *hand*$_1$ *the knife'*.
Since the merging of the subsequent actions has been decided, the parser performs another check:

- *if A'*$_{1b}$ *had a null object complement, then attribute a 'reference' feature to this null complement and bind it with the A'*$_{2b}$ *object-complement;*

This is a binding between the object that the first action is directed to and the object that is indeed affected by the subsequent action; in other words, the direction of the first action of the sequence (e.g. 'extend hand') functions as a deictic reference to something that becomes obvious when the subsequent action is executed (i.e. the object 'knife' in our example – see reference feature in the parse tree in figure 1). Such decision regarding the deictic nature of an action can only be taken, once the subsequent related action is found; i.e. the difference between 'extending a leg' and 'extending a leg towards X' is determined by the actions that follow these, such as 'extending another leg' (e.g. to stretch one's body) and 'kicking a ball' respectively.

The rules of the grammar may be applied recursively as actions combine in time sharing complements; for example, the '*grasp with hand*$_1$ *knife*' action shown in Figure 1, may be followed by a *'pin with knife apple'* action that extends the derived action structure further, adding one more constituent. This constituent is not just following the previous one in time, but its tool-complement is shared with the object-complement of the previous one. One can imagine infinite recursive applications of step 2 of our parser, e.g. adding one more constituent to our example, such as *'push with apple the plate'* (an action whose tool-complement is shared with the object-complement of the previous one). Recursion in step 2 is guided through the correlation of the tool-complement of subsequent actions; so, an action 'merges' with its tool (and optionally with an affected-object complement) and then 'merges' with a subsequent (or parallel) movement if they have the same tool-complement (e.g. *extend hand– grasp with hand X*), if the object complement of the preceding one is the tool complement of the one that follows (e.g. *grasp with hand knife – cut with knife bread*) or vice versa.

- *If the action that follows a thus far derived structure does not share a tool or object-complement with the preceding one, a new action substructure of A_{max} is created and processed applying the grammar rules bottom up.*

For example, the action sequence {extend $hand_1$ – enclose with $hand_1$ knife} may be followed with {extend $hand_2$}. In this case, no clues exist that this action forms part of the thus far derived structure and therefore a new action substructure in A_{max} is created (i.e. a second group of action-constituents) and parsing starts again applying rule 1.

Step 2 is applied until no more actions are available in the input visual/motoric stream. In this sense, A_{max} comprises an ordered set of action substructures, each substructure being an ordered set of actions itself.

In order for the parser to make the final decision regarding the merging of the action substructures, and thus, the derivation of one or more maximal action structures, two more steps are undertaken. These steps are the ones that lead to the application of rule 4 of the grammar, i.e. the attribution of the final goal to the maximal action structures:

STEP 3:

- *For each set A_{simple} of constituents of* $A_{max,}$ *which comprises only of the merging of an action terminal and its effector, find the first subsequent action-constituent set $A_{subsequent}$ whose first element F is not a simple action and shares the same tool-complement with the A_{simple} action. Expand A_{simple} adding a temporally combined to the A_{simple} unique constituent action trace linked to F; the object-complement of F is also bound with the null object-complement of the A_{simple} constituent through a reference feature sharing.*

[Figure 2]

This step is applied in cases such as the one depicted in Figure 2; there are two grasping actions (grasping a knife and grasping an apple) which take place in the following order: *'extend $hand_1$'*, '*extend $hand_2$*', '*enclose with $hand_2$ apple'*, '*enclose with $hand_1$ knife'*, and so on. The action 'enclose with $hand_1$knife' has a tool-complement that is not shared with the preceding action, but

it is shared with the first one. This is a case of a discontinuous action structure, a long-range dependency (see also section on action characteristics). This step of the action grammar implementation deals exactly with such phenomena, applying a *transformation*, on the derived action structure that allows a further *merging* of discontinuous actions (see Figure 3).

[Figure 3]

- *For each set $A_{complete}$ of constituents of* $A_{max,}$ *find the first subsequent action-constituent set $A_{subsequent}$ which comprises of at least one action structure with a tool-complement common with an object complement of a constituent of $A_{complete.}$ Link the two sets into the same complex action structure, though discontinuous in time;*

This case of step 3 deals with disruption phenomena between more complex structures than the ones presented above (cf. also examples in the next section).

Step 4:

- *Conclude with merging subsequent constituents of* A_{max}*, that share one or more object-complements.*

This last step is needed (a) for deciding whether actions that intervene (in terms of time sequence) between linked discontinuous action structures, belong to the same maximal action structure (see figure 4), and (b) for unifying action structures that share object complements at any distance, as usually the case in events with a loose structure (loose in terms of temporal sequence of constituents and presence/absence of some constituents), e.g. *rinse tomato, grasp knife, bring bowl, cut tomato with knife, pour oil into bowl* etc. for *preparing a salad*.

- *When no more merging can take place, check the goal feature of each maximal action structure (i.e. apply rule 4 of the grammar) and exit.*

[Figure 4]

Functions related to finding the tool of an action and the affected object are needed for the algorithm to work and are actually vital. As shown above, perceptual (e.g. visual) identification of body parts is a *sine qua non* requirement in this process and so is the notion of spatial intersection. The latter refers to a recursive 'merging' of body parts and objects as one comes into contact with

another. So, the tool of an action is any object that is either a moving body-part or a moving object spatially intersected with an effector and in synchrony with the effector. In the suggested algorithm, body parts and their intersection with other objects define not only the tools of an action, but also the affected objects, i.e. objects that are spatially intersected with a tool but they do not have the same motoric characteristics (e.g. one is not moving the other is, or they are both moving though not synchronized).

Note in the above that the tool complements of the action constituents of an action structure is an ordered set of entities, that cannot be empty (it is defining for any action); a body-part is the simplest tool, while its extension with other artifacts through a number of (tool-making) actions may form an infinitely complex tool.

Constraints on the use of body parts/effectors and their natural motors synergies should be incorporated in the algorithm sketched in this section, so that generation of correct and only correct action structures is guaranteed. The repertoire of possible motor synergies in human action should be taken into consideration. The suggested algorithm can deal with parallel syntax with slight modifications (e.g. checking not only for action constituents in a sequence, but in parallel timings too).

The algorithm can be used beyond human action to animal action. For non-biological actions, the 'tool' of the action is any natural force exerted on an object (e.g. a door closing because of the wind); employing methodologies for identifying such forces (e.g. the use of language for describing what is going on in a video) can lead to use of the above algorithm for analyzing non-biological actions too.

Action Grammar: Recursion, merge, move and the notion of tool-use

We have used a minimalist framework for defining a generative grammar of action; however, is such type of grammar really necessary? Wouldn't a regular grammar or a context free grammar be adequate for a formal analysis of the structure of action? In other words, is *recursion*, *merging* and *move*, all necessary for an action grammar?

Recursion is an important feature of generative grammars and has been shown to manifest itself in human language both:

(a) as tail recursion, a procedure that invokes another instance of itself as a final step, or in grammar terms, the embedding of a structure at the end of a structure of the same type [55] e.g. "*the man who knows your sister who works at the bookshop*". This is a complex noun phrase in which an anaphoric sentence ("who works…") is embedded at the end of another anaphoric sentence ("who knows…");

(b) as nested or true recursion, a procedure that invokes another instance of itself in mid-computation and then must resume the original procedure from where it left off, or in grammar terms, the embedding of a structure at the center of a structure of the same type [55] e.g. "*The cat the boy saw left*". This is a sentence, in which a sentence of the same type is embedded interrupting its structure, and thus, creating a discontinuous structure with long distance dependencies.

In the action grammar presented above, both types of recursion take place and they are both guided by the tool-use notion:

(a) tail recursion: this is recursion that takes place at step 2 of the algorithm presented above; it is the extension of an action with tool T and object O, with a following action with tool O on object X, which may be further extended with another action with tool X on object Y and so on; e.g.
"*extend hand$_1$ – grasp with hand$_1$ knife – cut with knife bread*", or in language terms:
"*extend hand$_1$, which grasps knife , which cuts bread*".

This is a complex action sequence, in which the third sub-action ("cut …") is embedded at the end of the second sub-action ("grasp…"). The role of the tool used in an action structure is vital in determining the recursion.

(b) true recursion: this type of recursion may appear in action sequences, in cases when one starts doing something before finishing off with something else; e.g.
"extend hand$_1$ – extend hand$_2$, grasp with hand$_2$ ball – grasp with hand$_1$ glass".

This is an action sequence, in which the "grasp glass" action is interrupted in its execution by another action ("grasp ball") of the same type; this results in a discontinuous structure and creates a long distance dependency between part of the "grasp glass" action. Such true recursion may manifest at action structures of a varying degree of complexity, i.e. at complex events such as preparing a salad:

"grasp with hand$_1$ knife – grasp with hand$_2$ cutting board, press with cutting board cloth - cut with knife bread", or in a more complex level:

"grasp with hand$_1$ knife, pin with knife bread – grasp with hand$_2$ fork, pin with fork cheese, lick with tongue cheese – bite with teeth bread".

In this case, the embedding disrupts the sequence of grasping a knife and actually using it, with an action structure of the same type. Thinking of everyday tasks, such embeddings seem quite frequent; of course, given that in the motor space some actions can take place in parallel, such embeddings are not always found in a neat sequence with the discontinuous elements, but rather part of them overlaps in time. Considering interaction with other people for performing a task, the phenomenon becomes even more frequent; actions of one agent "interrupt" those of another or seen from another perspective, one agent compensates for missing needs for completing a task before the other agent asks for them (e.g. brings a cutting board for cutting the bread) . It's a case of true coordination between actors.

Step 3 in the algorithm above addresses such discontinuous cases. Again, the notion of tool-use is employed for guiding the combination of discontinuous elements; it is the tool-complement of different actions that binds them together, though discontinuous in time.

Recursive *merging* in human language initiates a search for elements that solve variables in morpho-syntactic features of words/phrases; this is fundamental in the suggested action grammar too, because it guides all derivation. Through this search and merge mechanism actions combine with tools and affected objects and with other actions through unification of their features. Going beyond the merging of actions and corresponding tools/objects, merging in the action grammar takes place between elements of the same type too; this is what has been called in the minimalist framework 'internal merging' or 'move' [47]. Cf. for example, the '*extend hand to X*' sub-action in

Figure 1, in which there is clearly a reference to an object complement. The actual object complement forms part of a subsequent action. This is a probe-goal relation between the object-complement feature of one action and the corresponding feature of a subsequent one. Feature binding is common in action (due to object permanence), however, the reference mechanism manifests itself only in those cases that involve directed motion towards objects, but no contact with them. Furthermore, all 'disruption' cases (true recursion) mentioned above, are 'internal merging' cases too. Step 3 of the parsing algorithm makes use of the move operation leaving a trace at the position the constituent should normally be found and linking the trace with the constituent in its actual position in the action structure.

Based on all the above, we suggest that the generative grammar of action must necessarily allow for both tail and true recursion, and the use of merging and move operators. In such grammar, tool use plays an important role for the derivation of action structures.

Discussion

In employing a generative grammar for describing the structure of action, one substantiates experimental evidence on the common biological basis of language and action and feeds the long-standing debate on language evolution and what it is that makes human language unique [[56], [55, 57, 12]. However, why would one need a grammar of action and what does it mean, if tool use is indeed the computational structuring principle of action?

As shown above, a grammar of action has been sought for in both artificial intelligence and cognitive science; *generalisation, learning and prediction of action in both human cognition and computation* depends on identifying a structure of action that guides action related processing in both action recognition and action generation. Starting with artificial intelligence, event recognition and visual scene understanding have been the applications most interested in identifying a perceptual grammar of action for endowing robots and machines with the skills to recognize and interpret human behaviour. Large scale video processing depends on robust tools that perform visual object and visual action recognition; according to the suggested action grammar, recognition of human body parts [58](see Figures 4-5) is key to such applications and drives action recognition and in particular motor primitive recognition. Recognition of the spatial

intersection of body-parts and other objects is the next most important tool needed; this is technology that segments objects robustly [59] (see Figures 6-8) going beyond the visual merging of objects (e.g. the extension of a body part with the grasped object) and identifying not necessarily the type of object but instead its role as tool or object of interaction according to its spatial relation to a body part or an extended body part. Based on these two technologies, the action grammar can be used for parsing actions of any complexity, without ever going into full identification of the objects involved in these actions. All recognition is based on the pragmatic roles of the objects.

[Figure 5]

[Figure 6]

[Figure 7]

[Figure 8]

[Figure 9]

Going to action generation, robotics is interested in advanced motor control that allows an agent to plan the execution of an action (global control strategies) by combining motor primitives into actions that lead to attaining the final goal. In this task, putting actions in sequence and coordinating the use of the robot's effectors linearly or parallel in time for achieving a task is usually hard-coded and strictly dependent on the exact action that is to be executed. The suggested action grammar can be used as a sophisticated motor control planner that will generate correct and only correct sequences of actions depending on objects that the robot sees in its environment, experimenting with the different roles (tool or object of interaction) to be assigned to each object and with the execution of motor primitives; this is a guided object manipulation and exploration that can be used as a method by the robot to learn new behaviors, without necessarily being able to identify the exact type of objects. The grammar provides a way to determine the endpoint of a sequence of actions, without relying on knowledge of the exact action type.

As shown in the previous sections, the minimalist grammar of action comprises of features whose importance in an action grammar is corroborated by neurobiological evidence and so is the hierarchical and compositional nature of action structure. Action structures in our grammar are derived through *merging* which is a very basic operation, that of composition, and so is the move operation (since it is defined as 'internal merging'). However, *what drives* the merging in our

action grammar, calls for thorough exploration through experimentation. In the human action space, this question is in many ways equal to the question of what actually drives attention. In the minimalist framework, it is features that drive the merging. In our action grammar, it is indeed the tool and affected object complements that drive all merging, with the former playing a major role in all derivations.

In human cognition, there is a growing literature on the importance of the notion of tool use [13, 60]. However, no experiments have been reported on the role of this notion for structuring action. If our argument that 'tool-use' is the structuring principle of action has a biological basis (rather than merely a computational one), one would expect that an inability in humans to attribute the 'tool' role to an object within an action would be associated with inability to recognize or produce the hierarchical, recursive structure of an action.

Closer to the envisaged experiments are ones that show aphasics having problems in sequencing biological actions (e.g. to serve a cup of tea), while they have no problem in sequencing non-biological events (e.g. a bicycle falling) [14]. In this study, patients were also found to have severe problems in naming tools and tool-use, while they understood the global meaning of what they had seen. In a follow up of this work, it has been found that it is the ordering of *transitive and 'syntactic' biological actions* in particular that is affected by virtual lesions in the left Broca area 44 [61]. These actions involve hand-object interaction (i.e. tool-object interaction, e.g. cutting something as opposed to non-transitive ones such as 'getting up'), and have a compositional structure; they correspond to the ones that are derived through recursion in our action grammar.

So is it the attribution of the tool-role or the mechanism of recursion that is affected in such cases, or even both? Tool-use and language have been claimed to share computational mechanisms for processing complex hierarchical structures [60], a capacity that exists in primates with no language (of the complexity of human language) and which could have been exapted to support human grammatical ability [62]. Tool making in particular, has been speculated to have provided to action representation the capacity of recursion [13].

Through the minimalist grammar of action, we argue that action structure is recursive and it is tool-use that drives both merging (including move) and recursion.

Acknowledgements

Work reported in this paper is being funded by the POETICON Project Grant (FP7-ICT-215843), European Commission, Framework Program Seven. We would like to thank the POETICON consortium for our stimulating interaction and in particular Prof. Luciano Fadiga for inspiring discussions on the neuroscience of action.

References

[1] Lashley K. The problem of serial order in behaviour. In: Jefress L, editor. Cerebral mechanisms in behaviour. Wiley; 1951. .

[2] Leroi-Gourhan A. Le geste et la parole. Albin Michel; 1964. 2 volumes.

[3] Bauer P. Recalling past events: from infancy to early childhood. Annals of Child Development. 1995;11:25–71.

[4] Whiten A, Flynn E, Brown K, Lee T. Imitation of hierarchical action structure by young children. Developmental Science. 2006;9(6):574–582.

[5] Allen K, Ibara S, Seymour A, Cordova N, Botvinick M. Abstract structural representations of goal-directed behavior. Psychological science: a journal of the American Psychological Society. 2010;21.

[6] Fadiga L, Fogassi L, Gallese V, Rizzolatti G. Visuomotor neurons: ambiguity of the discharge or 'motor' perception? International Journal of Psychophysiology. 2000;35(2-3):165–177.

[7] Fogassi L, Ferrari P, Gesierich B, Rozzi S, Chersi F, Rizzolatti G. Parietal Lobe: from Action Organization to Intention Understanding. Science. 2005;308(5722):662–667.

[8] Aloimonos Y, Guerra-Filho G, Ogale A. The Language of Action: A New Tool for Human-Centric Interfaces. In: Aghajan H, Augusto J, Delgado R, editors. Human Centric Interfaces for Ambient Intelligence. Elsevier; 2009. p. 95–131.

[9] Aloimonos Y. HAL: Human Activity Language. Journal of Vision. 2008;8(6).

[10] Guerra-Filho G. A Sensory-Motor Linguistic Framework for Human Activity Understanding [Phd Thesis]. Department of Computer Science, University of Maryland, College Park; 2007.

[11] Chomsky N. The Minimalist Program. MIT Press; 1995.

[12] Chomsky N. Three Factors in Language Design. Linguistic Inquiry. 2005;36:1–22.

[13] Fadiga L, Craighero L, D'Ausilio A. Broca's area in language, action, and music. Annals of New York Academy of Science. 2009;1169:448–458.

[14] Fazio P, Cantagallo A, Craighero L, D'ausilio A, Roy A, Pozzo T, et al. Encoding of human action in Broca's area. Brain. 2009;132(7):1980–1988.

[15] Flash T, Hochner B. Motor Primitives in verterbrates and invertebrates. Current Opinion in Neurobiology. 2005;15:660–666.

[16] Viviani P. Do units of motor action really exist? In: Heuer H, Fromm C, editors. Generation and Modulation of Action Patterns. Springer Verlag; 1986. p. 201–216.

[17] Mussa-Ivaldi F, Bizzi E. Motor learning through the combination of primitives. Philosophical Transactions of the Royal Society - London B: Biological Sciences. 2009;355:1755–1769.

[18] Hart C, Giszter S. Modular premotor drives and unit bursts as primitives for frog motor behaviours. Journal of Neuroscience. 2004;24:5269–5282.

[19] Stein P. Neuronal control of turtle hind limb motor rhythms. Journal of Computational Physiology. 2005;191:213–229.

[20] Roitman A, Massaquoi S, Takahashi K, Ebner T. Kinematic analysis of manual tracking in monkeys: characterization of movement intermittencies during a circular tracking task. Journal of Neurophysiology. 2004;91:901–911.

[21] Pasalar S, Roitman A, Ebner T. Effects of speeds and force fields on submovements during circular manual tracking in humans. Experimental Brain Research. 2005;163:214–225.

[22] Fishbach A, Roy S, Bastianen C, Miller L, Houk J. Kinematic properties of on-line error corrections in the monkey. Experimental Brain Research. 2005;164:442–457.

[23] Jeannerod M. Object oriented action. Elserier and North Holland; 1994. p. 3–15.

[24] d'Avella A, Saltiel P, Bizzi E. Combinations of muscle synergies in the construction of a natural motor behaviour. Nature Neuroscience. 2003;6:300–308.

[25] Tresch M, Saltiel P, Bizzi E. The construction of movement by the spinal cord. Nature Neuroscience. 1999;2:162–167.

[26] Cheung V, d' Avella A, Tresch M, Bizzi E. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviours. Journal of Neuroscience. 2005;25:6419–6434.

[27] Ting L, MacPherson J. A limited set of muscle synergies for force control during a postural task. Journal of Neurophysiology. 2005;93:609–613.

[28] Ivanenko Y, Cappellini G, Dominici N, Poppele R, Lacquaniti F. Coordination of locomotion with voluntary movements in humans. Journal of Neuroscience. 2005;25:7238–7253.

[29] Arbib M. In: Shapiro S, editor. Schema Theory. Wiley Interscience; 1992. p. 1427–1443.

[30] Jeannerod M, Arbib M, Rizzolatti G, Sakata H. Grasping objects. The cortical mechanisms of visuomotor transformation. Trends in Neuroscience. 1995;18:314–320.

[31] Juhola M. A syntactic analysis method for eye movements of vestibule-ocular reflex. Computer Methods and Programs in Biomedicine. 1995;46:59–65.

[32] del Vecchio D, Murray R, Perona P. Decomposition of human motion into dynamics-based primitives with application to drawing tasks. Automatica. 2003;39:2085–2098.

[33] Jenkins O, Mataric M. Automated derivation of behavior vocabularies for autonomous humanoid motion. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems; 2003. p. 225–232.

[34] Rohrer B, Fasoli S, Krebs H, Hughes R, Volpe B, Frontera W, et al. Movement smoothness changes during stroke recovery. Journal of Neuroscience. 2002;22:8297–8304.

[35] Grinyagin I, Biryukova E, Maier M. Kinematic and dynamic synergies of human precision-grip movements. Journal of Neurophysiology. 2005;94:2284–2294.

[36] Kang N, Shinohara M, Zatsiorsky V, Latash M. Learning multifinger synergies: an uncontrolled manifold analysis. Experimental Brain Research. 2004;157:336–350.

[37] Jerde T, Flanders M. Coarticulation in fluent fingerspelling. Journal of Neuroscience. 2003;23:2383–2393.

[38] Fod A, Mataric M, Jenkins O. Automated derivation of primitives for movement classification. Autonomous Robots. 2002;12(1):39–54.

[39] Kahol K, Tripathi P, Panchanathan S. Automated gesture segmentation from dance sequences. In: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition; 2004. p. 883–888.

[40] Nakazawa A, Nakaoka S, Ikeuchi K, Yokoi K. Imitating human dance motions through motion structure analysis. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems; 2002. p. 2539–2544.

[41] Wang T, Shum H, Xu Y, Zheng N. Unsupervised analysis of human gestures. In: Proceedings of IEEE Pacific Rim Conference on Multimedia; 2001. p. 174–181.

[42] Ogale A, Karapurkar A, Aloimonos Y. View invariant modeling and recognition of human action using grammar. In: Lecture Notes in Computer Science. vol. 4358; 2007. p. 115–126.

[43] Chomsky N. Three Models for the Description of Language. IRE Transactions on Information Theory. 1956;2(3):113–124.

[44] Chomsky N. Syntactic Structures. Mouton de Gruyter; 1957.

[45] Chomsky N. Aspects of the Theory of Syntax. MIT; 1965.

[46] Chomsky N. Lectures on Government and Binding: The Pisa Lectures. Mouton de Gruyter; 1993.

[47] Lasnik H. The minimalist program in syntax. Trends in Cognitive Sciences. 2002;6(10):432–437.

[48] Iriki A, Sakura O. The neuroscience of primate intellectual evolution: natural selection and passive and intentional niche construction. Philosophical Transactions of the Royal Society:B. 2008;363:2229–2241.

[49] Iriki A, Tanaka M, Iwamura Y. Coding of modified body schema during tool use by macaque postcentralneurones. Neuroreport. 1996;14:2325–2330.

[50] Mantovani G, Bufalari I, doΩ½Ausilio A, Fadiga L. The brain representation of objects and tools. POETICON Project; 2011. D3.2b.

[51] Luria A. The Working Brain. Penguin; 1973.

[52] Cangelosi A, Metta G, Sagerer G, Nolfi S, Nehaniv C, Fischer K, et al. Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics. IEEE Transactions on Autonomous Mental Development. 2010;2(3):167–195.

[53] Cattaneo L, Fabbri-Destro M, Boria S, Pieraccini C, Monti A, Cossu G, et al. Impairment of action chains in autism and its possible role in intention understanding. Proceedings of the National Academy of Sciences. 2007;104(45):17825–30.

[54] Li Y, Aloimonos Y. The Joint Synergies: Partitioning Human MoCap Data into Action Segments. In: Proceedings of the Robot Learning Workshop, Robotics: Science and Systems; 2009.

[55] Pinker S, Jackendoff R. The Faculty of Language: What's special about it? Cognition. 2005;95:201–236.

[56] Hauser M, Chomsky N, Fitch W. The Faculty of Language: What Is It, Who Has It, and How Did It Evolve? Science. 2002;298:1569–1579.

[57] Fitch W, Hauser M, Chomsky N. The evolution of the language faculty: Clarifications and Implications. Cognition. 2005;97:179–210.

[58] Summerstay D, Aloimonos Y. Learning to recognize using anisotropic kernel. In: Proceedings of the Biologically Inspired Cognitive Architectures Workshop; 2010.

[59] Mishra A, Aloimonos Y. Active Segmentation. International Journal of Humanoid Robotics. 2009;6(3):361–386.

[60] Stout D, Toth N, Schick K, Chaminade T. Neural correlates of Early Stone Age toolmaking: technology, language and cognition in human evolution. Philosophical Transactions of the Royal Society - London B: Biological Sciences. 2008;363(1499):1939–1949.

[61] Clerget E, Winderickx A, Fadiga L, Olivier E. Role of Broca's area in encoding sequential human actions: a virtual lesion study. Neuroreport. 2009;20(16):1496–1499.

[62] Higuchi S, Chaminade T, Imamizu H, Kawato M. Shared neural correlates for language and tool use in Broca's area. Neuroreport. 2007;20(15):1376–1381.

Figure Captions:

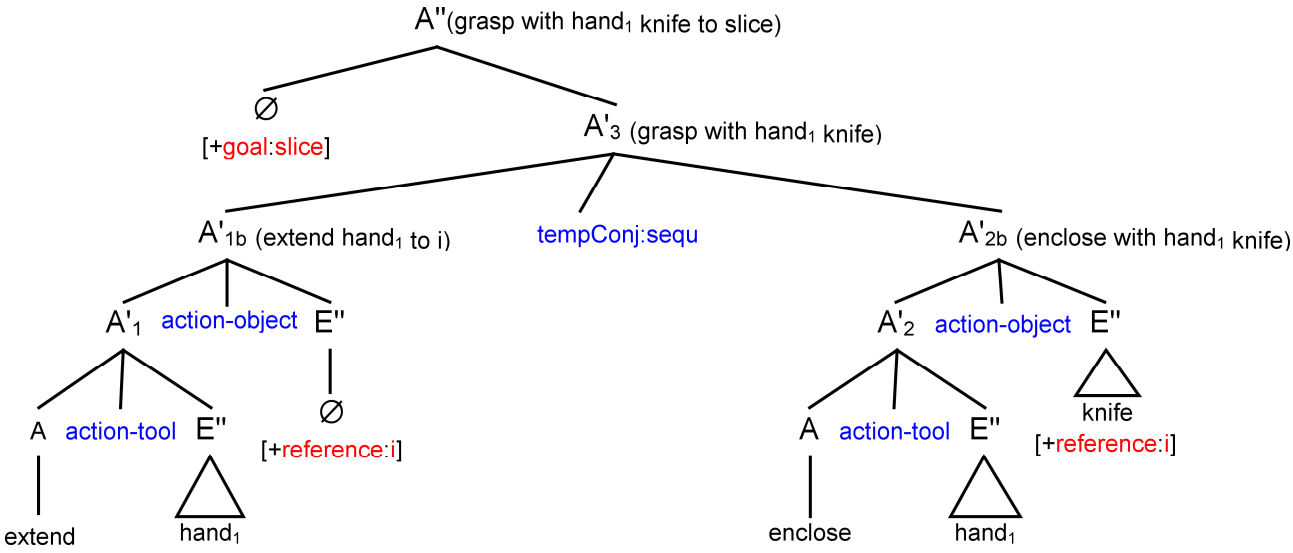

Figure 1: Action parse tree for "grasp with hand$_1$ knife to slice"; A stands for action primitives (terminals), A' for action structures (non-terminals), A'' for the maximal projection of an action structure. E'' is the maximal projection of an entity structure. Triangles in the tree denote that the corresponding part of the tree is not fully analysed for keeping the figure simple. Parentheses present morphological features of the corresponding tree nodes, in an 'attribute:value' format; the plus sign denotes the presence of such features, a minus would denote the absence of a feature. The exact type of relation between branches of the tree is clearly denoted for clarification purposes; 'action-tool' and 'action-object' are complements of an action and as such they are inherently related to the corresponding action structure. Sub-actions of a complex action are sequential or parallel in time, i.e. they are related through the corresponding 'temporal conjunction' type (tempConj:sequ, or tempConj:par).

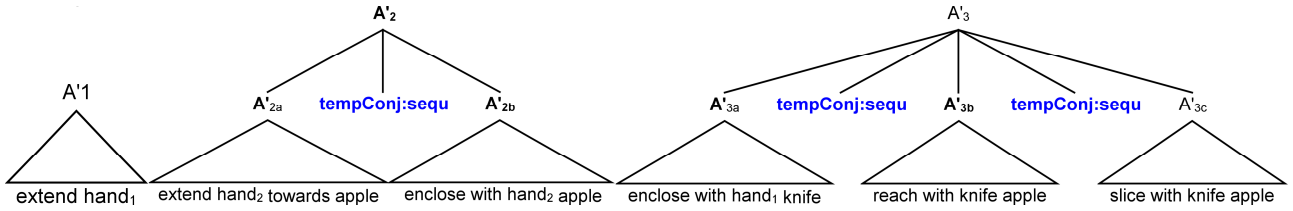

Figure 2: Incomplete parsing of the sequence: 'extend hand$_1$', 'extend hand$_2$', 'enclose with hand$_2$ apple', 'enclose with hand$_1$ knife', 'reach with knife apple' and 'slice with knife apple'. After step 2 of the action parser, three stand-alone action structures are derived rather than one structure comprising all three of them with the final goal of slicing.

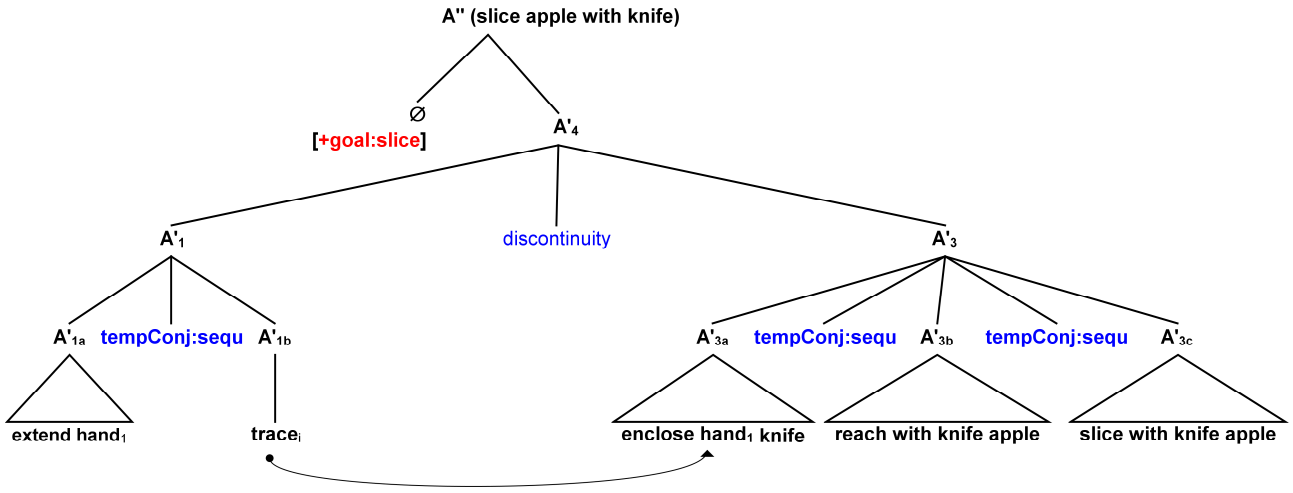

Figure 3: Action parse tree of the structure after applying step 3, i.e. after the move operation has been applied. A'$_{3a}$ shares the same tool complement with A'$_{1a}$ (hand$_1$) and its object-complement (knife) is referred to by A'$_{1a}$. Therefore, its expected

position is semantically exactly after $A'_{1a.}$ in position $A'_{1b.}$ *However, due to the disruption by other actions, this position is empty; the action is in position* $A'_{3a.}$ *Thus, a 'trace' of the action is left in position* A'_{1b} *which is linked with the action in position* $A'_{3a.}$ *The two structures* A'_1 *and* A'_3 *are not temporally combined, they are discontinuous; the actions that intervene in between may or may not be part of the same action structure. This is what step 4 checks.*

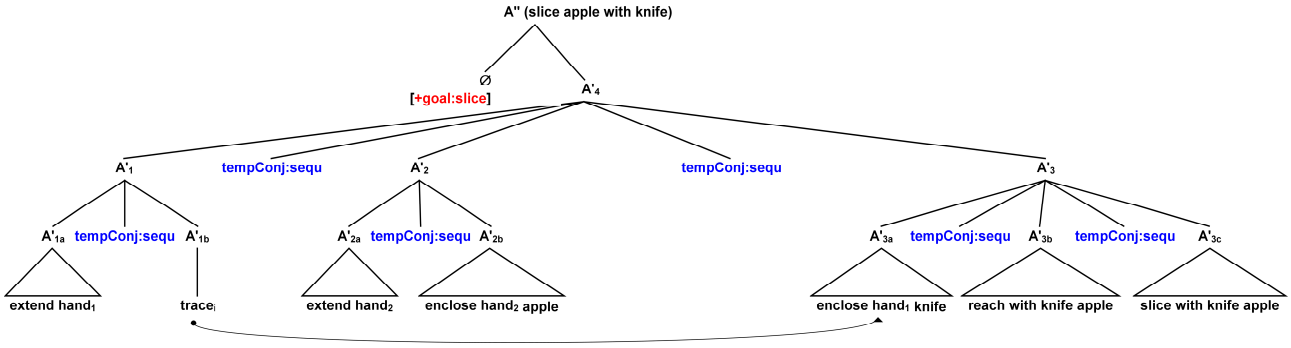

Figure 4: The maximal action structure for the sequence: 'extend $hand_1$*, grasp with* $hand_2$ *apple, grasp with* $hand_1$ *knife, reach with knife apple, slice with knife apple'. After linking the discontinuous sub-action constituents in step 3 of the parser, a decision is taken regarding the action structure* A'_2 *that intervenes temporally causing the disruption: in applying step 4 of the parser, structure* A'_2 *is found to share an object-complement with constituents of the action structure* A'_3 *(i.e. the apple). This is enough for considering* A'_2 *to be a constituent of the more complex action structure* A'_4*. Thus, the three independent action structures presented in figure 2 which were not combined into a common structure due to a disruption phenomenon are now all linked into a sequence with a common final goal.*

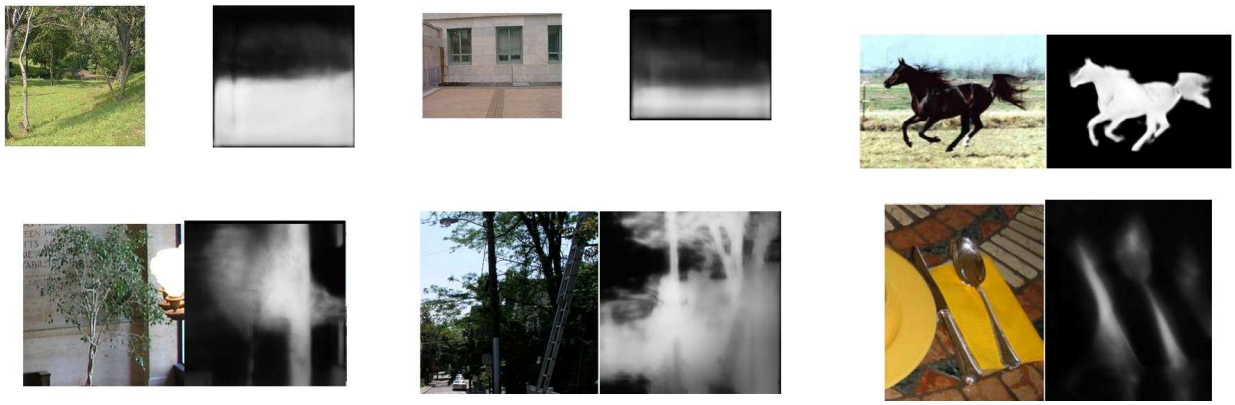

Figure 5: Using techniques from machine learning one can develop new non linear filters that when applied to an image produce a new image, where the intensity of a pixel is proportional to the probability that the pixel lies inside the image of a specific category, e.g. silverware. On the left is the image, on the right the output of the filter.

Figure 6: Just as we can learn filters for objects, we can also learn them for body parts, legs, arms, heads, torsos, hands. On the left is the image; on the right is the output of filters for body parts denoted in different colors.

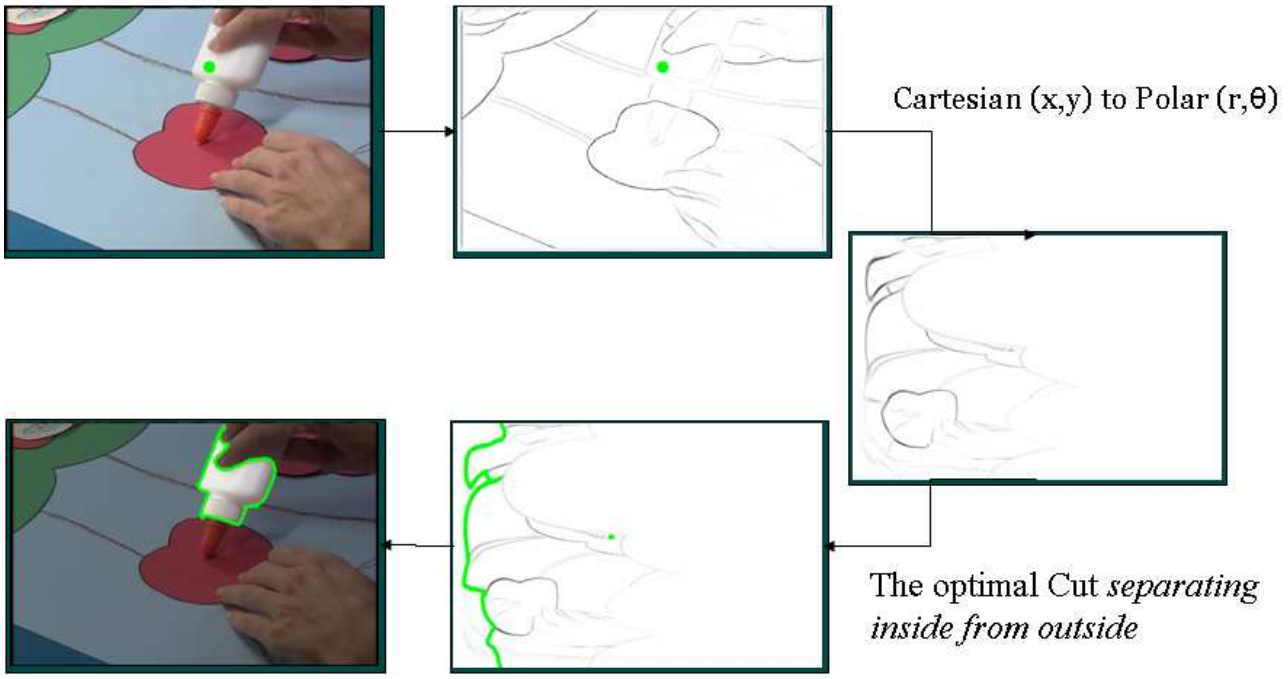

Figure 7: By fixating at a part of a scene (selecting a point in the image) we can segment the object containing the fixation point, in this case a glue bottle from a hands and crafts activity. Images and video courtesy of Johns Hopkins Workshop on Vision and Language.

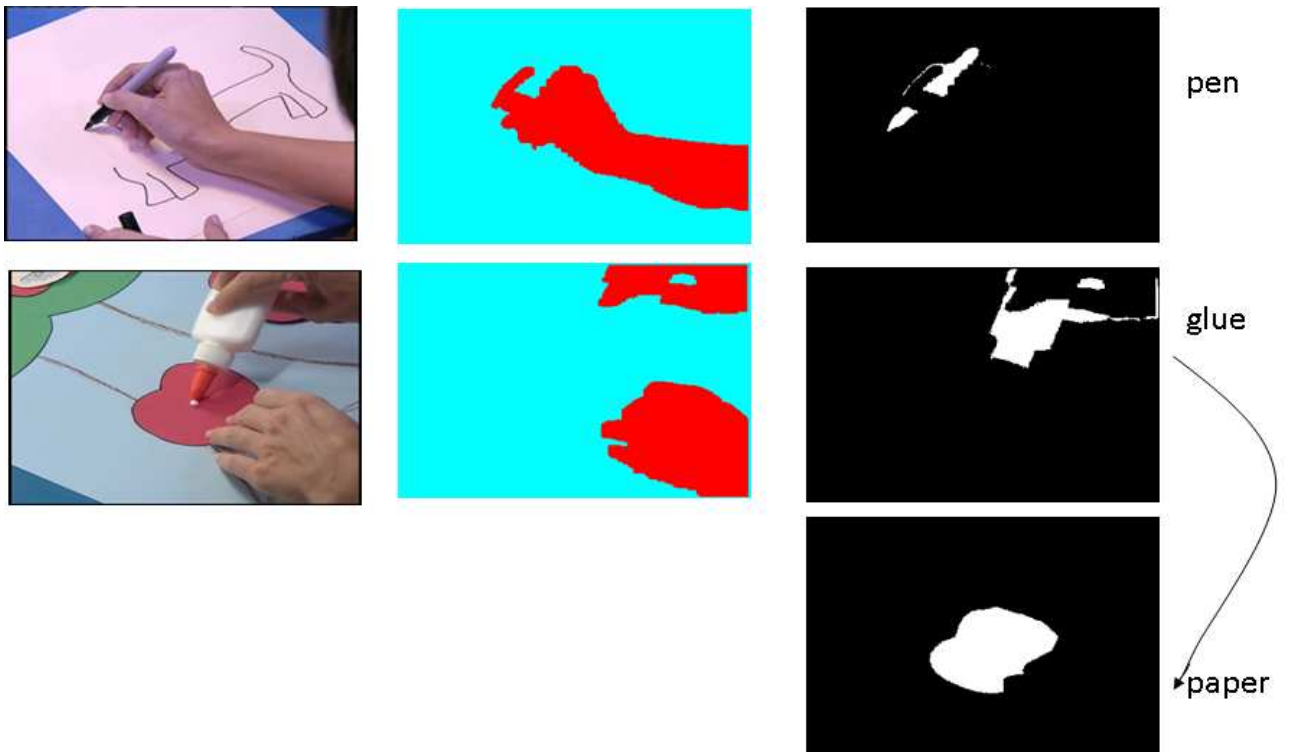

Figure 8: By using the filters described in figures 5 and 6, we can process videos of human activity to segments hands, tools and objects participating in actions. Top row: Left: image from a "drawing" activity. Middle: segmentation of hands; Right: segmentation of the object in the hand (pen). Bottom row shows results from another activity. Images and video courtesy of Johns Hopkins Workshop on Vision and Language.

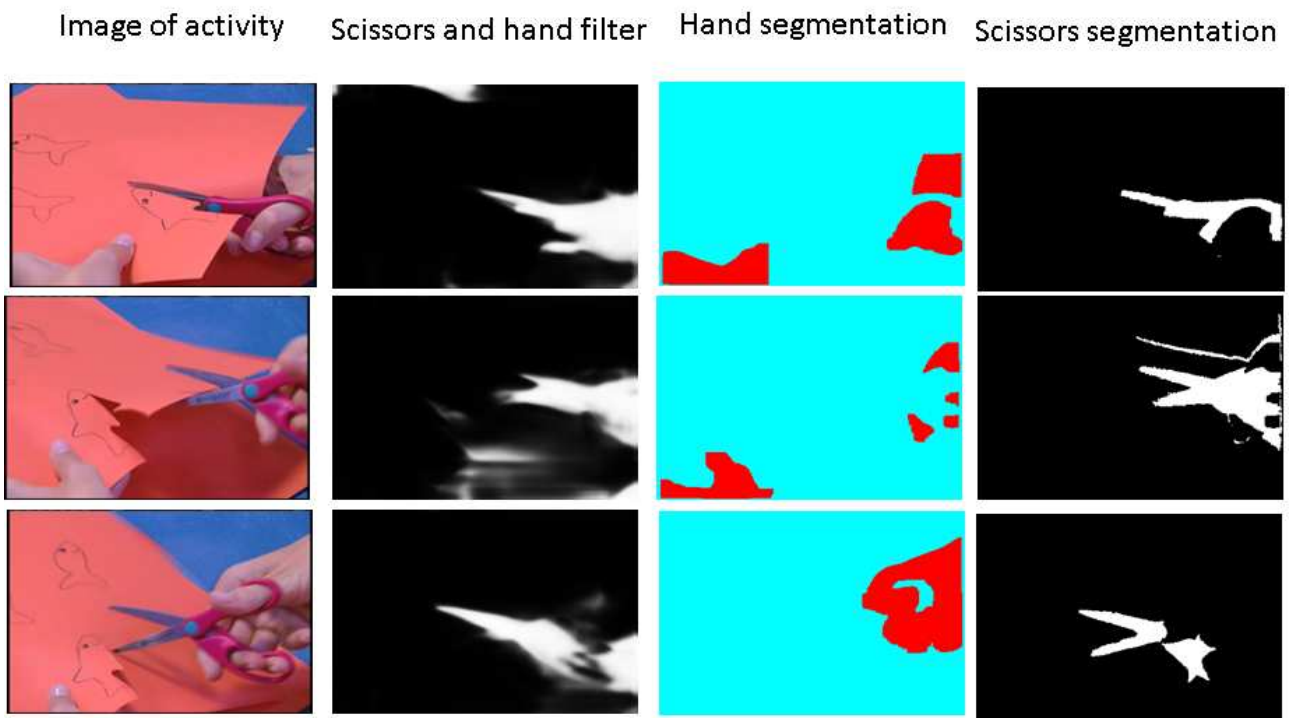

Figure 9: Image of the activity (cutting paper with scissors), scissors and hand filter output, hand segmentation, scissors segmentation using the technique of figure 5 and the filter output. Images and video courtesy of Johns Hopkins Workshop on Vision and Language.

View publication stats