Body Posture Recognition as a Discovery Problem: a Semantic-based Framework*

Michele Ruta[1], Floriano Scioscia[1], Maria di Summa[2], Saverio Ieva[1], Eugenio Di Sciascio[1], and Marco Sacco[3]

[1]Politecnico di Bari, Bari, Italy
{michele.ruta, floriano.scioscia, saverio.ieva, eugenio.disciascio}@poliba.it
[2] Consiglio Nazionale delle Ricerche, Bari, Italy
maria.disumma@itia.cnr.it
[3] Consiglio Nazionale delle Ricerche, Milano, Italy
marco.sacco@itia.cnr.it

Abstract. The automatic detection of human activities requires large computational resources to increase recognition performances and sophisticated capturing devices to produce accurate results. Anyway, often innovative analysis methods applied to data extracted by off-the-shelf detection peripherals can return acceptable outcomes. In this paper a framework is proposed for automated posture recognition, exploiting depth data provided by a commercial tracking device. The detection problem is handled as a semantic-based resource discovery. A simple yet general data model and a corresponding ontology create the needed terminological substratum for an automatic posture annotation via standard Semantic Web languages. Hence, a logic-based matchmaking allows to compare retrieved annotations with standard posture descriptions stored as individuals in a proper Knowledge Base. Finally, non-standard inferences and a similarity-based ranking support the discovery of the best matching posture. This framework has been implemented in a prototypical tool and preliminary experimental tests have been carried out w.r.t. a reference dataset.

Keywords: Action recognition, Resource Discovery, Semantic-based matchmaking, Ubiquitous Computing

1 Introduction

The detection of articulate activities has been studied for a long time, mainly focusing research on video analysis. Nevertheless, recent technological enhancements opened the way for novel possibilities. Infrared depth sensors allow to discern three-dimensional shapes in an environment, a kind of information which

* The authors wish to acknowledge support from National Operative Program project *Res Novae* (Grid, Building and Road Objectives for Environment and Energy).

is often hard to derive from standard video data. Unfortunately, until latest years depth sensors were very expensive and therefore they were used in limited applications and circumstances. More recently, following some product and process evolutions, several low-cost multi-sensor devices become commercially available, as for example *Microsoft Kinect*.[1] It is equipped with a standard RGB video camera, a microphone and an infrared depth sensor with resolution and accuracy enough for several practical applications. It must be also considered that deficiencies in capture precision (particularly in general-purpose use cases, where performance decreases due to variety and generality of the input), could be counterbalanced by novel software-side analyses often profiting from the large availability of data corpuses.

In this paper a framework is proposed for an automated posture detection, exploiting depth data provided by the *Microsoft Kinect* tracking device. A recognition problem is handled as a resource discovery one, grounded on a semantic-based matchmaking [1]. The needed terminology (*a.k.a.*, ontology) for geometry-based semantic descriptions of postures has been encapsulated in a Knowledge Base (KB) also including several instances representing poses templates to be detected. Skeleton model data retrieved by the Kinect are pre-processed on-the-fly to identify *key postures*, *i.e.*, unambiguous and not transient body positions (which typically correspond to the early or the final state of a gesture). Each key posture is then annotated adopting standard Semantic Web languages based on the Description Logics (DL) formalism [2]. Hence, non-standard inferences allows to compare the retrieved annotations with templates populating the KB and a similarity-based ranking supports the discovery of the best matching posture. The theoretical framework has been implemented in a prototype and several experiments have been carried out w.r.t. a public dataset [3]. Preliminary results report a satisfactory recognition precision for various kinds of postures, validating the feasibility and effectiveness of the proposed approach.

The remainder of the paper is organized as follows. Most relevant related work is surveyed in Section 2, the theoretical framework and the proposed approach are presented in Section 3 while details about designed prototype along with the related evaluation are in Section 4. Conclusion and perspectives in Section 5 terminate the paper.

2 Related Work

After the needed preparatory steps on data extracted by capturing devices, specific recognition algorithms can be divided in *machine learning* and *ontology* based ones. Approaches based on machine learning theory can be either supervised or unsupervised. In *supervised* techniques, collected data is divided into a training set and a test set, in order to train the recognition algorithm on the former and evaluate its performance on the latter. A limit of that kind of approaches is that they require a relatively large corpus of labeled data to be

[1] http://www.microsoft.com/en-us/kinectforwindows/

built for training, usually by hand. Furthermore, the resulting models achieve good accuracy only for the specific scenarios they are thought for. They are not reusable and scalable when individual behavior or environmental conditions change. Hence, the recognition of a large diversity of activities in real-world application scenarios could be deemed as impractical. *Unsupervised* methods try to construct recognition models directly from unlabeled data, by manually assigning a probability to each possible activity and using a graph-based, algebraic or probabilistic model. The limit of the unsupervised learning methods lies in the assignment of probabilistic parameters. *Semi-supervised* learning [4] has recently received significant attention as a technique to balance system accuracy and required human and computational effort. It combines small-scale expert labeled data and large-scale unlabeled data based on certain assumptions.

Ontology-based activity recognition follows a completely different approach. It exploits a logic-based knowledge representation for activity and sensor data modeling, and logical reasoning to perform activity detection. Such approaches: (i) use a semantically rich formalism to explicitly define a library of models for all possible instances in a domain; (ii) aggregate and translate sensed data in logical formulae grounded on the above terminological box; (iii) perform reasoning to infer a minimal model based on the set of observed actions. Ontology-based approaches bridge the semantic gap between low-level observations and high-level detected phenomena. In order to exploit this benefit, in [5] a video movement ontology was engineered to allow automatic annotation of human movements in the classic Benesh notation. A standard ontology-based framework for video annotation is presented in [6], allowing a hierarchical representation of events, by means of Video Event Representation Language (VERL) and Video Event Markup Language (VEML). The description of complex events is built by aggregating elementary concepts relating them by means of temporal relationships. However, VERL is rather complex and verbose, so that exhaustive definition of recognition rules is not practical for large sets without domain-specific customizations and/or user-friendly tools. Automated analysis of surveillance video is one of the most frequent applications of ontology-based activity recognition and annotation [6–9]. Chen and Nugent [10] proposed an ontology-based approach which is more similar to the one described here: an Activities of Daily Living (ADL) DL ontology was produced for activity modeling and reasoning in the context of smart homes. Subsumption is used to enable a flexible activity recognition at different levels of detail, depending on the amount of knowledge acquired from the environment. However, as pointed out in [11], classical DL inference services are not enough in these cases, since recognition/interpretation tasks cannot be trivially considered as *classification* ones, but they are more similar to *model construction*. The main weakness of logic-based approaches is their general inability to represent vagueness and uncertainty. This issue could be partially solved by exploiting fuzzy DLs. Furthermore, most of ontology-based approaches offer no mechanism for deciding whether one particular model is more effective than another. Finally, they have adopted a top-down approach so far, focusing only on high-level activities and events.

3 Framework and Approach

The framework proposed here can be considered as a part of the gesture recognition technique based on key posture detection [12]. Each recognition process occurs in three steps: (i) *posture description*, which provides posture annotations; (ii) *posture detection*, which sequentially identifies a few reference poses, namely *key postures*; (iii) *gesture identification*, which labels recognized gestures from sequences of key postures. The present work focuses on the first two stages. Data capture is provided by the popular Kinect platform. Particularly, the NUI (Natural User Interface) API, provided in the Kinect for Windows SDK,[2] uses the depth data stream to detect human presence in the infrared sensor range: at most two people can be recognized and tracked simultaneously. For each of them, the NUI produces a human body model, named *skeleton*, composed by 20 joints. Each joint point is defined by its *(x, y, z)* coordinates, expressed w.r.t. a Cartesian spatial reference system whose origin is located on the depth sensor itself. NUI can also mark a point as "inferred" if its coordinates are not directly detected by the sensor (*e.g.*, the body part is occluded by another object), but are estimated via proprietary algorithms by the processing unit embedded in the Kinect device. Inferred joint data are commonly affected by significant noise.

3.1 Architecture

The architecture of the proposed system is depicted in Fig. 1. It is based on three main components:

1. **Posture annotator**, which exploits skeleton tracking capabilities, in order to give a description of body pose with unambiguous semantics. A proper domain ontology has been developed to this aim, as described in the next section.
2. **Posture repository**, storing key postures to be recognized as instances in a Knowledge Base –expressed w.r.t. the shared reference ontology;
3. **Semantic matchmaking engine**, exploiting non-standard logic-based reasoning to support approximated matches, key posture ranking and explanation of outcomes.

3.2 Skeleton Representation

Due to software-side correction effort, a considerable accuracy in detection is not needed –if compared to on-screen rendering– hence a straightforward joint-angle skeleton model is adopted. It provides invariance to sensor orientation and skeleton variations among different individuals. A less approximated representation could improve the posture labeling process, but it would introduce not negligible technical issues: (i) heavier processing; (ii) lack of robustness of the annotation procedure w.r.t. detection errors from joint position data obtained by Kinect sensor; (iii) more complex semantic matchmaking, adversely affecting precision and recall of key posture recognition. The body postures are

[2] http://www.microsoft.com/en-us/kinectforwindows/develop/

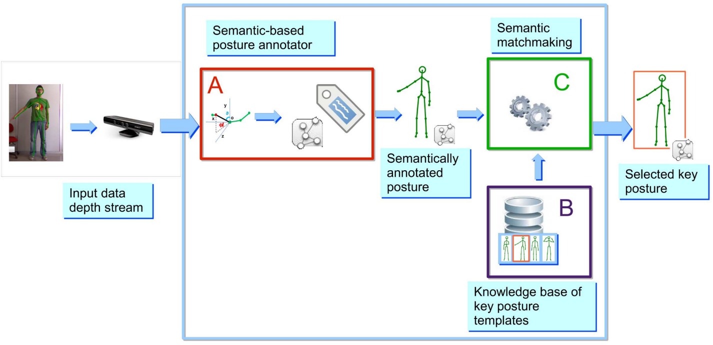
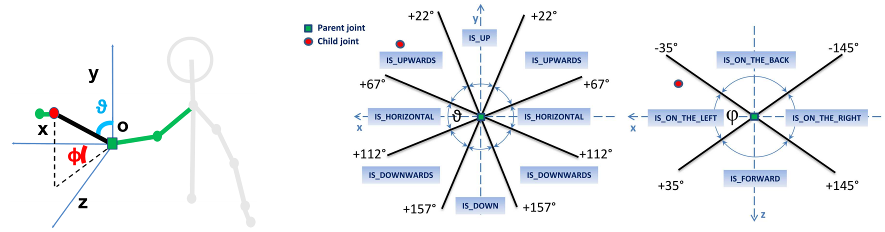

Semantic-based posture annotator
Semantic matchmaking
A
C
B
Input data depth stream
Semantically annotated posture
Selected key posture
Knowledge base of key posture templates

Fig. 1. Architectural block diagram of the proposed framework

basically determined by the mutual position of bones (for example a leg is bent or stretched depending on status of femur and tibia). The proposed framework adopts a posture description model similar to the one in [13], by converting each joint position, defined w.r.t. the Cartesian coordinate system of the NUI API, to a local spherical system. The new reference system keeps x and y axes parallel to the former ones, while z (depth) axis is opposed, *i.e.*, it points toward the Kinect sensor, as illustrated in Fig. 2a. The origin is locally and progressively translated to the "parent" joint, according to the hierarchical order defined by the Kinect NUI API. Following this model, each skeleton segment is represented by zenith and azimuth angles $\{\theta, \varphi\}$; the radius is ignored because the length of each bone is fixed for a given subject. The proposed model omits to annotate body extremities (feet and hands) because they are often inferred by the NUI API, so the posture detection process could be affected by some inaccuracy. In spite of its simplistic nature, this model has been chosen because it allows to represent a broad variety of human postures, also keeping under control the complexity of both recognition and annotation automatic procedures. Such raw angular information are labeled using the Cone-Shaped Directional (CSD) logic framework [14] as formal reference. Particularly, in the proposed model a set of labeled directions is used for given θ and φ values between each parent-child joint couple. This results in a series of cone-shaped 3D regions, as illustrated in Fig. 2b. Regions are defined to conform to qualitative intuition or common-sense knowledge, *e.g.*, for φ the back and forward regions are wider than the ones on the sides.

3.3 Semantic-based Posture Annotation

In order to enable a fully automated posture annotation as well as the further matchmaking for recognition, the skeleton representation model described above must be translated using an ontology language grounded on a given logic and provided with a proper semantics. A prototypical ontology modeling the domain



(a) Spherical coordinates reference system **(b)** The set of Cone Shaped Directional relations adopted for θ and φ in the proposed model

Fig. 2. Proposed model

of interest has been defined, using a subset of *OWL 2*[3] elements corresponding to the $\mathcal{ALN}$ (Attributive Language with unqualified Number restrictions) formal language of DLs family. An ontology is a formal conceptualization of the problem domain. Relevant entities like the human body parts and the steps of the recognition process (posture, gesture, action) can be modeled explicitly with unambiguous meaning. Furthermore, this knowledge can be shared among researchers, developers and practitioners, allowing the latter to extend the core model in order to meet requirements of their specific use cases. Basically, an ontology is composed of: (i) *classes* (*a.k.a.*, concepts), denoting types of objects; (ii) *properties* (*a.k.a.*, roles), representing relationships either between pairs of objects as classes instances (*object properties*) or between class instances and data-oriented attributes for which a data type is provided (*datatype properties*, *a.k.a.*, features on concrete domains). These basic elements are used to build concept expressions by exploiting logical constructors; each language of the DLs family is characterized by a given set of allowed constructors, which affects algorithmic complexity of inference procedures. Main patterns of the $\mathcal{ALN}$ ontology designed for the purposes of this work are reported hereafter, adopting the example in Fig. 3.

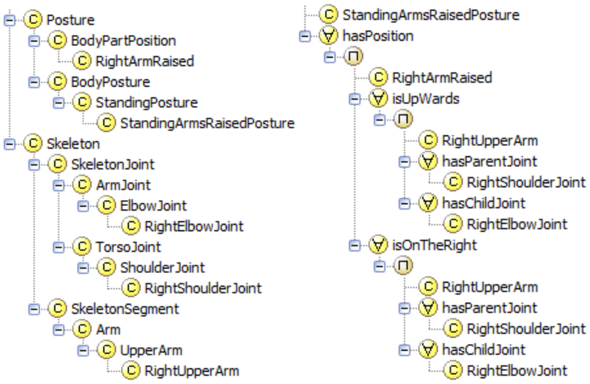
– Joints are modeled as subclasses of the `SkeletonJoint` class. Likewise, skeleton segments are expressed as subclasses of `SkeletonSegment`. Each segment is related to the joints at its extremities through `hasParentJoint` and `hasChild Joint` properties.

– Skeleton body part positions are expressed by means of subclasses of the `Skeleton BodyPart` element, modeling common body part poses (*e.g.*, `RightArmRaised`). Each configuration is related to a subclass of `SkeletonSegment` through azimuth and zenith properties. The mapping between $\{\theta, \varphi\}$ values and the object properties is achieved via the CSD framework, as described above.

– A set of classes representing pre-defined body postures have been modeled as subclasses of `BodyPosture`. They are related to the above-mentioned body part position classes through the `hasPosition` property. As an example, Fig. 3 de-

[3] OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation, 11 December 2012, http://www.w3.org/TR/owl2-overview/

picts only a portion of `StandingArmsRaisedPosture` definition, which describes the right arm; remaining body parts are described following the same modeling pattern.



(a) Class hierarchy **(b)** Key posture template

Fig. 3. Posture ontology model

The proposed approach offers many benefits. Practitioners (*e.g.*, physiotherapists) will be able to extend easily the provided ontology with their domain-specific knowledge, also in a collaborative way. The high level of abstraction allows changing the assumptions underlying an implementation if the knowledge about the domain changes. Hard-coding assumptions in programming languages makes them not only hard to find and understand but also hard to change, in particular for someone without programming expertise. In addition, explicit specifications of domain knowledge are useful for new users who must learn the meaning of the terms in the domain. Possible extensions of the recognition framework to include gestures and actions detection simply lead to an incremental expansion of the domain ontology accordingly.

3.4 Semantic-based Key Posture Recognition

The proposed approach is devoted to use technologies borrowed from the Semantic Web initiative and deductive inferences to solve the posture detection problem: (i) by exploiting machine-understandable annotated descriptions; (ii) by reasoning on obtained expressions inferring implicit knowledge from annotations; (iii) by adopting the Open World Assumption (OWA), saying that the lack of a feature in a resource representation should not be necessarily interpreted as a constraint of absence. Such a general modeling framework allows users to create Knowledge Bases of annotated posture templates, suitable for different scenarios. Then, the key posture identification is managed as a semantic-based resource discovery on the KB, where logic-based inference services provide a similarity ranking which can be seen as a confidence degree in the identification.

Also a detailed semantic-based explanation of results is returned as useful outcome. *Semantic matchmaking* can be defined as the process of finding the best matches among n resources $S_i (i = 1, \ldots n)$ w.r.t. a given request R, where both request and resources are annotated w.r.t. a common reference ontology [1]. In the proposed approach, S_i are the key posture templates in the KB, while R is the current annotated posture. Most reasoners usually provide two standard *Satisfiability* and *Subsumption* inference services for matchmaking. In particular, *Subsumption* returns *true* iff all features requested in R are provided by S_i, but full matches are infrequent in practical scenarios, so it usually gives hopeless 'no match' results. *Concept Abduction* (CA) non-standard inference service, originally formalized and applied in e-commerce scenarios [1], is adopted in this work, allowing to: (i) provide an outcomes explanation beyond the trivial "yes/no answer of subsumption tests and (ii) enable a logic-based relevance ranking of a set of available resources w.r.t. a given query. If R and S_i are compatible –*i.e.*, not contradictory– but S_i does not fully satisfy R, CA allows to determine what is missing in S_i in order to completely satisfy R. The solution H (for *Hypothesis*) to CA represents "why" the subsumption relation does not hold. H can be interpreted as *what is requested in* R *and not specified in* S_i. In this way, it is possible to support non-exact matches and to define metrics upon H to compute logic-based ranking of resources best approximating the request [1]. Given S and R in Conjunctive Normal Form [1], Algorithm 1 (reported later on) finds a minimal solution for CA in $\mathcal{ALN}$ DL w.r.t. the number of conjuncts in H and computes the corresponding *penalty function* [1] for S w.r.t. R. A toy example should clarify the above process. A semantic-based request of a *person standing up with straight and parallel legs, left arm straight along left side, right arm pointing downward to the right, head up* can be formally expressed as:

(**R**)**DetectedPosture** $\equiv$ $Skeleton \sqcap \forall\ hasPosition.(\forall\ isUp.(Head \sqcap SpinalColumnSegment) \sqcap \forall\ isDownWards.(RightUpperArm \sqcap RightLowerArm) \sqcap \forall\ isOnTheRight.(RightUpperArm \sqcap RightLowerArm) \sqcap \forall\ isDown.(LeftUpperArm \sqcap LeftLowerArm \sqcap LeftUpperLeg \sqcap LeftLowerLeg \sqcap RightUpperLeg \sqcap RightLowerLeg))$.

Let us consider the following resources in the key posture templates KB.
Standup with right arm outstretched (S_1)*: person standing up with straight and parallel legs, left arm straight along left side and right arm outstretched, head up looking straight ahead.* W.r.t. domain ontology, it is expressed as:

($\mathbf{S_1}$) $\equiv$ $StandupPosture \sqcap \forall\ hasPosition.(HeadUp \sqcap LeftArmAlongSide \sqcap RightArmOutstretched))$.

Standup with raised arms on side (S_2)*: person standing up with straight and parallel legs, left arm raised on left side and right arm raised on right side, head up looking straight ahead.* In DL notation:

($\mathbf{S_2}$) $\equiv$ $StandupPosture \sqcap \forall\ hasPosition.(HeadUp \sqcap LeftArmRaised \sqcap RightArmRaised))$.

It can be noticed that the structure of the proposed ontology allows to keep posture annotations short and easy to understand, because details are encapsulated in the definition of referenced classes. **StandupPosture** $\equiv$ $BodyPosture \sqcap \forall\ hasPosition.(RightLegStraight \sqcap LeftLegStraight)$.

RightLegStraight $\equiv$ $\forall\ isDown.(RightUpperLeg \sqcap RightLowerLeg)$.

LeftLegStraight $\equiv$ $\forall\ isDown.(LeftUpperLeg \sqcap LeftLowerLeg)$.

RightArmOutstretched $\equiv \forall\ isHorizontal.(RightLowerArm) \sqcap \forall\ isDownwards.(RightUpperArm) \sqcap \forall\ isOnTheRight.(RightUpperArm \sqcap RightLowerArm).$

RightArmRaised $\equiv \forall\ isUpwards.(RightLowerArm) \sqcap \forall\ isHorizontal.(RightUpperArm) \sqcap \forall\ isOnTheRight.(RightUpperArm \sqcap RightLowerArm).$

LeftArmRaised $\equiv \forall\ isUpwards.(LeftLowerArm) \sqcap \forall\ isHorizontal.(LeftUpperArm) \sqcap \forall\ isOnTheLeft.(LeftLowerArm \sqcap LeftUpperArm).$

LeftArmAlongSide $\equiv \forall\ isDown.(LeftUpperArm \sqcap LeftLowerArm).$

RightLowerLeg $\sqsubseteq \forall\ hasParentJoint.(RightKneeJoint) \sqcap \forall\ hasChildJoint.(RightAnkleJoint).$

LeftUpperArm $\sqsubseteq \forall\ hasParentJoint.(LeftShoulderJoint) \sqcap \forall\ hasChildjoint(LeftElbowJoint).$

RightUpperArm $\sqsubseteq \forall\ hasParentJoint.(RightShoulderJoint) \sqcap \forall\ hasChildjoint(RightElbowJoint).$

When an annotated posture (request) is received, the following processing steps are performed.

1. Stored key postures (resources) are extracted from the repository.
2. The reasoning engine computes CA with Algorithm 1 between request and each resource.
3. Results of semantic matchmaking are transferred to the utility function calculation module, which computes the final ranking according to the scoring function reported afterwards.
4. Finally, the ranked list of best resource records is returned. Furthermore, a similarity threshold is introduced: resources key posture templates having an overall score w.r.t. the request worse than this threshold will not give back. Otherwise, detected posture(s) are provided along with their scores and H values, which justify the outcome. By solving the Concept Abduction Problem it is possible to compute the missing features of the key posture templates S_i, needed to reach a full match with R. In particular, in S_1 *RightLowerArm* should be in horizontal position. On the other hand, in S_2 *RightUpperArm* should be slightly lowered and $LeftLower\ Arm$ and *LeftUpperArm* should be down. In formulae:

$\mathbf{H_{R,S_1}} \equiv \forall\ isDownwards.(RightLowerArm).$

$\mathbf{H_{R,S_2}} \equiv \forall\ isDown.(LeftUpperArm \sqcap LeftLowerArm) \sqcap \forall\ isDownwards.(RightUpperArm).$

In the proposed approach, semantic similarity is computed via the following *utility function*:

$$f\left(R,S\right) = 100 * \left[1 - \frac{penalty\left(R,S\right)}{penalty\left(R,\top\right)}\right] \qquad (1)$$

where *penalty* measures the CA-induced semantic distance between real-time annotation R and key posture template instance S; this value is normalized dividing by the distance between R and the universal concept −*Top* or *Thing*− which depends only on axioms in the ontology and is the maximum possible value. In the example, the overall similarity score is 91% for S_1 and 60% for S_2.

4 Prototype and Experiments

As a proof of concept, the proposed framework has been implemented in a software prototype, extending the existing *Kinect Toolbox*.[4] Thanks to the GUI in Fig. 4, the tool enables users to compose semantic annotations for body postures,

[4] Kinect Toolbox, `http://kinecttoolbox.codeplex.com/`

Algorithm: $abduce\,(\langle \mathcal{L}, D, S, \mathcal{T} \rangle)$

```
Require: $\langle \mathcal{L}, R, S, \mathcal{T} \rangle$ with $\mathcal{L} = \mathcal{ALN}$, acyclic $\mathcal{T}$
Ensure: $\langle H, penalty \rangle$ with $penalty \geq 0$ and
    $H \in \mathcal{ALN}$
$H := \top$;
$penalty := 0$;
for all concept name $A$ in $R$ do
    if no $B$ in $S$ exists s.t. $B \sqsubseteq A$ then
        $H := H \sqcap A$;
        $penalty := penalty + 1$;
    end if
end for
for all concept $(\geq\ x\ P)$ in $R$ do
    if $(\geq\ y\ P)$ exists in $S$ and $y < x$ then
        $H := H \sqcap (\geq\ x\ R)$;
        $penalty := penalty + \frac{x-y}{x}$;
    else if no $(\geq\ y\ P)$ exists in $S$ then
        $H := H \sqcap (\geq\ x\ P)$;
        $penalty := penalty + 1$;
    end if
end for
for all concept $(\leq\ x\ P)$ in $R$ do
    if $(\leq\ y\ P)$ exists in $S$ and $x < y$ then
        $H := H \sqcap (\leq\ x\ P)$;
        $penalty := penalty + \frac{y-x}{x}$;
    else if no $(\leq\ y\ P)$ exists in $S$ then
        $H := H \sqcap (\leq\ x\ P)$;
        $penalty := penalty + 1$;
    end if
end for
for all concept $\forall P.E$ in $D$ do
    if $\forall P.F$ exists in $S$ then
        $\langle H', penalty' \rangle := abduce\,(\langle \mathcal{L}, E, F, \mathcal{T} \rangle)$;
        $H := H \sqcap \forall P.H'$;
        $penalty := penalty + penalty'$;
    else
        $H := H \sqcap \forall P.E$;
        $penalty := penalty + 1$;
    end if
end for
return $\langle H, penalty \rangle$
```

Algorithm 1: Concept Abduction in $\mathcal{ALN}$ DL

without requiring specific knowledge of Semantic Web languages and underlying logic-based formalisms. In literature many tools aim to support developers, *e.g.*, *KINA* toolkit [15] and *DejaVu* [16]. Conversely, the goal of the proposed system is to allow users, who are typically practitioners and not necessarily developers, to build a set of representative postures for a given domain by composing visually a high-level description. A typical usage experience consists of the following interaction steps:

– **Data capture:** input streams include depth and RGB data provided by Kinect in real-time. When a subject is facing the Kinect sensor, her/his movements are tracked and skeleton data are retrieved and displayed on the panel (A). The automatic annotation engine described in the previous section calculates body segment angles and builds the corresponding semantic description. The system allows also to process pre-recorded data, in the form of skeleton frame sequences.

– **Annotation:** if the user presses the 'Annotate Posture' button on panel (D), it is shown the annotation of the just captured posture. Panel (B) provides an intuitive tree-like graphical representation. The above described ontology is loaded and its elements populate the upper part of the panel: (i) classes (*e.g.*, `SkeletonSegment`, `BodyPart Position`, etc.); (ii) object properties (*e.g.*, *hasPosition* links a `Body Posture` to a `BodyPartPosition`); (iii) datatype properties.[5] The current posture annotation is displayed in the lower portion of panel (B). The user can edit it through drag-and-drop of classes and properties from the ontology: context menus appear whenever additional information should be specified. Then s/he can save it in the KB by clicking on the 'Apply' button just

[5] This kind of constructors is not used in the current version of the ontology, but may be used in future extensions to deal with gestures and actions.

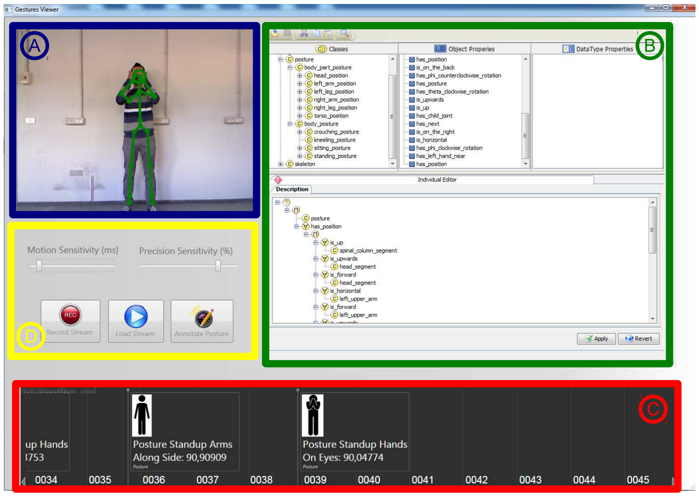


Fig. 4. Prototype tool screenshot

below the annotation.

– **Semantic matchmaking:** the current posture is detected as a key posture only if it lasts for a tunable *motion sensitivity* parameter. In that case, it is automatically processed for recognition. An embedded lightweight reasoner [17] is exploited to perform semantic matchmaking between the real-time annotated posture and each key posture instance stored in the KB. The result is a list of recognized key postures ordered w.r.t. the overall compatibility score. The recognized key posture is the one that best matches the real-time annotation, and it is added to the list (C).

Basic key postures can be slightly modified or even enriched to help domain experts in fitting requirements of a specific scenario. Expert personnel can customize posture descriptions through the above interface to improve the recognition capabilities. . For example, in telerehabilitation scenarios, a physiotherapist records for a patient the postures to avoid in front of the Kinect sensor. Then he loads the data, annotates the postures and saves them in the KB. From then on, every time the patient takes one of these bad postures, the system automatically recognizes and registers it and the physical therapist can figure out the patients compliance remotely. An experimental campaign was carried out to obtain a preliminary performance evaluation of the proposed approach. Postures were selected from a subset of gestures collected in [3]. Each gesture was repeated at least 10 times consecutively. The sequences were recorded after giving instructions to the subjects using different formats (images, video and/or text). Preliminarily, by exploiting the developed tool, a set of annotated key postures were defined. Then the prototype tool was tuned to a motion sensitivity of 0.3 sec -equivalent to 9 frames at the default NUI API sampling frequency of 30 frames/sec- and a precision threshold of 60%. Results are reported in Table 1 and refer to seven key postures belonging to a subset of five gestures taken from the dataset. For each key posture the following data were measured: (i) relative

frequency of detected occurrences N, divided by the total number of real executions observed in each sequence n; (ii) average semantic score $\overline{f(R,S)}$; (iii) standard deviation $\sigma_{f(R,S)}$. Outcomes show that the proposed method is reliable, having 88.0% rate of correct posture recognition, particularly if the subject is scrupulous in the execution of the task. Looking in greater detail, the following remarks can be made:
– *Start music/raise volume*: results are good except for the *Sequence2*, where the achieved recognition rate is lower because the subject did not raise arms enough in some cases.
– *Crouch or hide* and *Take a bow*: even if key postures are always identified correctly, low average scores were obtained, probably because of skeleton tracking instability and occlusion issues in the input data.
– *Navigate to next menu*: recognition frequencies are not very high for some sequences, while semantic scores are always high, except for *Sequence5.* That case is anomalous because the gesture was performed with the wrong hand; in fact, the system detected the symmetric posture in some cases.
– *Put on night vision goggles*: obtained results are very good in terms of frequency and semantic score; only *Sequence2* did not achieve a perfect recognition rate, because of some skeleton tracking instability in the input data.

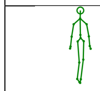
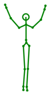
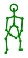
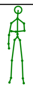
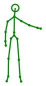
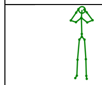
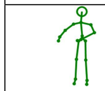
5 Conclusion

The paper introduced a general-purpose framework and approach for semantic-based posture annotation and recognition. It exploits 3D skeleton joint position data provided by a Microsoft Kinect device as input. A general model was devised to characterize body parts and most common poses, and an ontology was designed to give them the needed formal terminology through standard Semantic Web languages. The posture recognition problem has been basically handled as resource discovery via semantic matchmaking, exploiting non-standard inference services from managing approximate matching. The theoretical framework has been implemented in a prototypical tool devoted to prove the effectiveness of the proposed approach: results obtained w.r.t. a reference dataset provide a promising proof of concept.
Future work aims to enhance the presented framework toward gesture recognition. This goal will require an extension of both data model and domain ontology allowing to annotate a gesture as ordered sequence of postures. Also the semantic matchmaking framework will be extended to support a more articulate recognition. Finally, a broader experimentation and comparison with state-of-the-art approaches is planned.

References

1. Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., Tinelli, E.: A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces. International Journal of Electronic Commerce **12**(2) (2007) 127–154

Table 1. Experiments result

	Sequence1			Sequence2			Sequence3			Sequence4			Sequence5		
KP	**N/n**	$\overline{f(R,S)}$	$\sigma_{f(R,S)}$	**N/n**	$\overline{f(R,S)}$	$\sigma_{f(R,S)}$	**N/n**	$\overline{f(R,S)}$	$\sigma_{f(R,S)}$	**N/n**	$\overline{f(R,S)}$	$\sigma_{f(R,S)}$	**N/n**	$\overline{f(R,S)}$	$\sigma_{f(R,S)}$
Start music/raise volume															
	11/11	100	0	13/13	100	0	12/12	100	0	11/11	100	0	12/12	100	0
	8/10	96.14	6,91	6/11	98,53	2,65	11/11	97,61	5,05	10/10	86,11	3,08	11/11	98,71	1,26
Crouch or hide															
	10/10	80,43	2,46	10/10	73,89	4,57	10/10	76,14	5,36	10/10	80,97	3,10	10/10	84,60	2,25
Navigate to next menu															
	6/10	90,41	4,83	6/12	93,05	3,58	14/16	87,39	0	12/12	88,17	2,58	4/10	67,26	3,62
	6/10	94,46	2,49	7/11	94,94	3,24	16/16	94,92	2,52	11/11	94,87	1,50	0/10	-	-
Put on night vision goggles to change the game mode															
	10/10	91,59	2,08	9/10	73,87	11,72	10/10	90,15	1,17	11/11	93,79	2,09	11/11	89,88	1,97
Take a bow															
	11/11	72,84	2,59	10/10	71,73	0,52	10/10	69,66	0,74	8/10	71,67	1,75	10/10	69,50	3,61

2. Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Cambridge University Press (2002)
3. Fothergill, S., Mentis, H., Kohli, P., Nowozin, S.: Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, ACM (2012) 1737–1746
4. Zhang, T., Xu, C., Zhu, G., Liu, S., Lu, H.: A generic framework for video annotation via semi-supervised learning. Multimedia, IEEE Transactions on **14**(4) (Aug.) 1206–1219
5. Saad, S., De Beul, D., Mahmoudi, S., Manneback, P.: An ontology for video human movement representation based on benesh notation. In: Multimedia Computing and Systems (ICMCS), 2012 International Conference on, IEEE (2012) 77–82
6. François, A., Nevatia, R., Hobbs, J., Bolles, R., Smith, J.: VERL: an ontology framework for representing and annotating video events. MultiMedia, IEEE **12**(4) (2005) 76–86
7. Vrusias, B., Makris, D., Renno, J.P., Newbold, N., Ahmad, K., Jones, G.: A framework for ontology enriched semantic annotation of CCTV video. In: Image Analysis for Multimedia Interactive Services, 2007. WIAMIS'07. Eighth International Workshop on, IEEE (2007) 5–5
8. Akdemir, U., Turaga, P., Chellappa, R.: An ontology based approach for activity recognition from video. In: Proceedings of the 16th ACM international conference on Multimedia. MM '08, New York, NY, USA, ACM (2008) 709–712
9. SanMiguel, J.C., Martinez, J.M., Garcia, A.: An ontology for event detection and its application in surveillance video. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE (2009) 220–225
10. Chen, L., Nugent, C.: Ontology-based activity recognition in intelligent pervasive environments. International Journal of Web Information Systems **5**(4) (2009) 410–430
11. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Ontology-based context representation and reasoning for object tracking and scene interpretation in video. Expert Syst. Appl. **38**(6) (June 2011) 7494–7510
12. Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A.W., Campos, M.F.M.: Real-time gesture recognition from depth data through key poses learning and decision forests. In: Sibgrapi 2012 (XXV Conference on Graphics, Patterns and Images), Ouro Preto, MG, IEEE (august 2012)
13. Raptis, M., Kirovski, D., Hoppe, H.: Real-time classification of dance gestures from skeleton animation. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, New York, NY, USA, ACM (2011) 147–156
14. Renz, J., Mitra, D.: Qualitative direction calculi with arbitrary granularity. In: In Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence, Springer (2004) 65–74
15. Reis, B., Teixeira, J.a.M., Breyer, F., Vasconcelos, L.A., Cavalcanti, A., Ferreira, A., Kelner, J.: Increasing Kinect application development productivity by an enhanced hardware abstraction. In: 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, New York, NY, USA, ACM (2012) 5–14
16. Kato, J., McDirmid, S., Cao, X.: Dejavu: Integrated support for developing interactive camera-based programs. In: 25th Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, ACM (2012) 189–196
17. Ruta, M., Scioscia, F., Di Sciascio, E., Gramegna, F., Loseto, G.: Mini-ME: the Mini Matchmaking Engine. In Horrocks, I., Yatskevich, M., Jimenez-Ruiz, E., eds.: OWL Reasoner Evaluation Workshop (ORE 2012). Volume 858 of CEUR Workshop Proceedings., CEUR-WS (2012) 52–63