Semantic matchmaking for Kinect-based posture and gesture recognition

Michele Ruta*, Floriano Scioscia*, Maria di Summa†, Saverio Ieva*, Eugenio Di Sciascio* and Marco Sacco‡
* Politecnico di Bari
via Orabona 4,
I-70125, Bari, Italy
Email: (michele.ruta, floriano.scioscia, saverio.ieva, eugenio.disciascio)@poliba.it
† Consiglio Nazionale delle Ricerche
via Lembo 38F,
I-70124, Bari, Italy
Email: maria.disumma@itia.cnr.it
‡ Consiglio Nazionale delle Ricerche
via Bassini 15,
I-20133, Milano, Italy
Email: marco.sacco@itia.cnr.it

***Abstract*—Innovative analysis methods applied to data extracted by off-the-shelf peripherals can provide useful results in activity recognition without requiring large computational resources. In this paper a framework is proposed for automated posture and gesture recognition, exploiting depth data provided by a commercial tracking device. The detection problem is handled as a semantic-based resource discovery. A general data model and the corresponding ontology provide the formal underpinning for automatic posture and gesture annotation via standard Semantic Web languages. Hence, a logic-based matchmaking, exploiting non-standard inference services, allows to: (i) detect postures via on-the-fly comparison of the retrieved annotations with standard posture descriptions stored as instances of a proper Knowledge Base; (ii) compare subsequent postures in order to recognize gestures. The framework has been implemented in a prototypical tool and experimental tests have been carried out on a reference dataset. Preliminary results indicate the feasibility of the proposed approach.**

I. Introduction and motivation

Recent technological enhancements opened the way for novel possibilities in activity recognition. Infrared depth sensors allow to distinguish three-dimensional (3D) shapes in an environment, a kind of information which is often hard to derive from standard video data [1]. Unfortunately, until latest years depth sensors were very expensive and therefore they were used in limited applications and circumstances. Nevertheless, nowadays low-cost multi-sensor devices, as for example *Microsoft Kinect*, have become available. They are generally equipped with a standard RGB video camera, a microphone and an infrared depth sensor with resolution and accuracy enough for practical applications. It has to be considered that deficiencies of capture precision (especially in general-purpose use cases, where performance decreases due to variety of the input) could be compensated by novel software-side analysis approaches. Particularly, studies in machine learning techniques, algorithms and tools have enabled novel classes of data interpretation approaches. In addition, the exploitation of logic-based and approximate discovery strategies leverage non-exact matching results to counterbalance possible weaknesses in capturing activities.

In this paper a framework is proposed for automated posture and gesture detection, exploiting depth data provided by the *Microsoft Kinect* tracking device. Introduced novel features are: (i) adoption of standard Semantic Web technologies for posture and gesture annotations; (ii) exploitation of non-standard inferences services provided by an embedded matchmaker [2] to automatically detect postures and gestures. Particularly, the recognition problem is handled as a resource discovery, grounded on a semantic-based matchmaking [3]. 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 pose templates to be detected. 3D body model data detected by Kinect are pre-processed on-the-fly to identify *key postures*, *i.e.*, unambiguous and not transient body positions. They typically correspond to the initial or final state of a gesture. Each key posture is then annotated adopting standard Semantic Web languages based on the Description Logics (DL) formalism. Hence, non-standard inferences allows to compare the retrieved annotations with templates populating the Knowledge Base and a similarity-based ranking supports the discovery of the best matching posture.

The theoretical framework has been implemented in a prototype and experiments have been carried out on a public dataset [4]. Preliminary results report a satisfactory recognition precision for various kinds gestures, validating the feasibility and effectiveness of the proposed approach.

The remainder of the paper is organized as follows. The theoretical framework and the proposed approach are presented in Section II while details about designed prototype are in Section III. Most relevant related work is surveyed in Section IV before conclusion and future remarks in Section V.

II. Proposed approach

The proposed framework carries out gesture recognition in three steps: (i) *posture description*, which provides posture annotations; (ii) *posture detection*, which sequentially identifies a few reference postures, named *key postures*; (iii) *gesture identification*, which labels recognized gestures as sequences of key postures. Data capture is provided by the Kinect for Windows SDK[1] which uses the depth sensor to produce a 3D human body model (*skeleton*) composed by 20 joints having *(x, y, z)* coordinates.

The architecture of the proposed system is depicted in Figure 1. It is based on three main components:
(A) ***posture annotator***, which exploits skeleton tracking in order to give a description of body poses with unambiguous semantics.
(B) ***semantic matchmaking engine*** [2] exploiting non-standard logic-based reasoning to support approximated discovery and ranking of key postures along with the explanation of outcomes. (C,D) ***posture and gesture repositories***, storing instances in a Knowledge Base, expressed according to the shared reference ontology.

A. Data model

The proposed framework adopts a posture description model similar to the one in [5], by converting each joint position originally in Cartesian coordinates to a local spherical reference system (see Figure 2). Each skeleton segment is referenced via its zenith and azimuth angles $\{\theta, \varphi\}$ with respect to the parent joint. Thanks 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 enough. It provides invariance to sensor orientation and skeleton variations among different subjects and, in spite of its simplistic nature, it allows to represent a broad variety of human postures keeping under control the complexity of the automatic procedures both for annotation and recognition. Raw angular information is labeled using the Cone-Shaped Directional (CSD) logic framework [6] as formal reference. Particularly, in the proposed model a set of labeled directions is used for given θ and φ value ranges between each parent-child joint pair. A series of cone-shaped 3D regions are so defined. The proposed model does not annotate spherical coordinates of body extremities (feet and hands), because they are often inferred by the Kinect SDK, so the posture detection process could be affected by some inaccuracy. In order to annotate hand location, proximity to other joints (*e.g.*, other hand or head) is evaluated.

B. Ontology for postures and gestures

In order to enable a fully automated gesture annotation from a sequence of body postures 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 of interest has been defined, using a subset of *OWL 2*[2] elements corresponding to the $\mathcal{ALN}$ (Attributive Language with unqualified Number restrictions) formal language of DLs family. DLs are based on *concepts* representing classes of objects, *roles* joining pairs of objects and *individuals* which are specific named objects. $\mathcal{ALN}$ includes the Top ($\top$) and Bottom ($\bot$) concepts and the following constructors: atomic concept negation $\neg$, conjunction $\sqcap$, qualified universal restriction $\forall$, unqualified existential restriction $\exists$ and unqualified number restrictions $\geq, \leq$. Main patterns are reported hereafter.

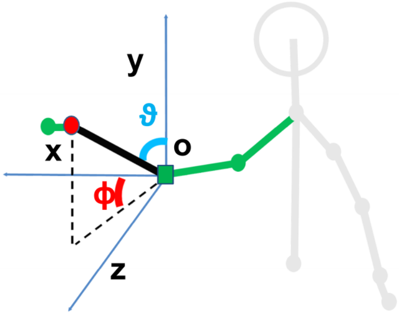


Fig. 2: Spherical coordinates reference system.

Body elements. Joints are modeled as subclasses of the `SkeletonJoint` class. Likewise, skeleton segments are expressed as subclasses of `SkeletonSegment` (Figure 3a). Each segment is related to the joints at its extremities through `hasParentJoint` and `hasChildJoint` properties (*e.g.*, in Figure 3b).

Body part positions. Skeleton body part positions are expressed by means of subclasses of `SkeletonBodyPart` 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 described above.

Hands. Hands position proximity to other joints is expressed through `hasRightHandNear` and `hasLeftHandNear` properties. Furthermore, cardinality restrictions are also considered.

Postures. Subclasses of `BodyPosture` represent pre-defined body poses. They are related to body part position classes through the `hasPosition` property. As an example, Figure 3b depicts a portion of `StandingHandsOnEyesPosture` definition, which describes the right arm; remaining body parts are described following the same modeling pattern.

Simple gestures. Gestures are detected via the spatial position of body parts, which can move together in order to shape a higher-level movement (*e.g.*, raising both arms).

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

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

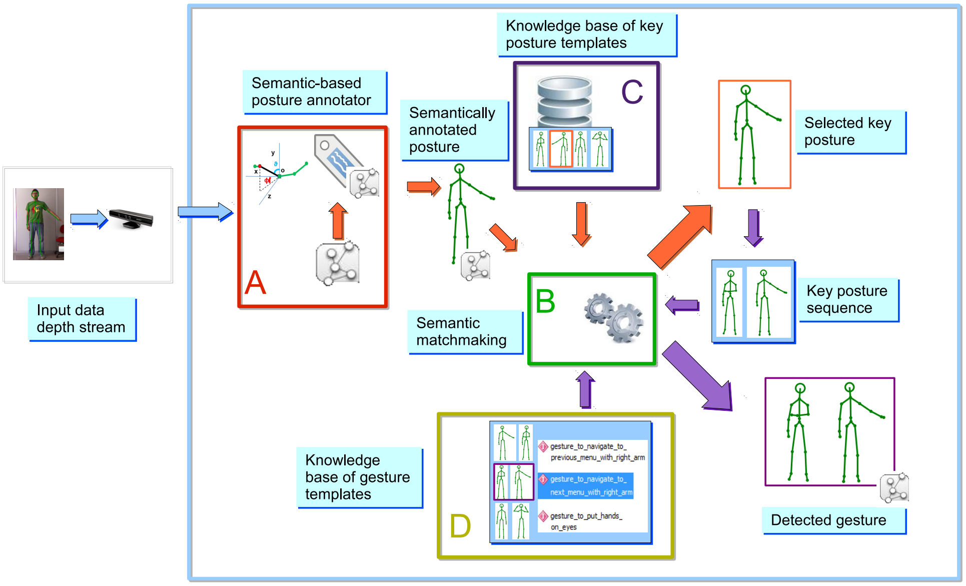

Knowledge base of key posture templates
Semantic-based posture annotator
Semantically annotated posture
C
Selected key posture
Input data depth stream
A
B
Semantic matchmaking
Key posture sequence
Knowledge base of gesture templates
D
gesture_to_navigate_to_previous_menu_with_right_arm
gesture_to_navigate_to_next_menu_with_right_arm
gesture_to_put_hands_on_eyes
Detected gesture

Fig. 1: Architectural block diagram of the proposed framework

Particularly, gestures involving a single body part are expressed as subclasses of `SimpleGesture`. Moreover, `SimpleGesture` can be divided into two kinds of movements: (i) `LimbGesture`, *i.e.*, gestures related to a whole body part (*e.g.*, raising right arm on side); (ii) `SegmentGesture`, which model more articulate movements involving body extremities (*e.g.*, waving hand to say hello). In both cases, each gesture class is labeled as a sequence of body postures, where the preceding one is related to the following one through the `hasNext` property. For instance, `BringingRightArmForwardGesture` class definition is shown in Figure 3.

Complex gestures. Complex gestures involving more body parts are expressed as subclasses of `CompositeGesture`. They are related to `SimpleGesture` through the `hasGesture` property. As an example, Figure 3 shows `ShootingGesture` definition, which is composed of `BringingRightArmForward` and `BringingLeftArmForward`.

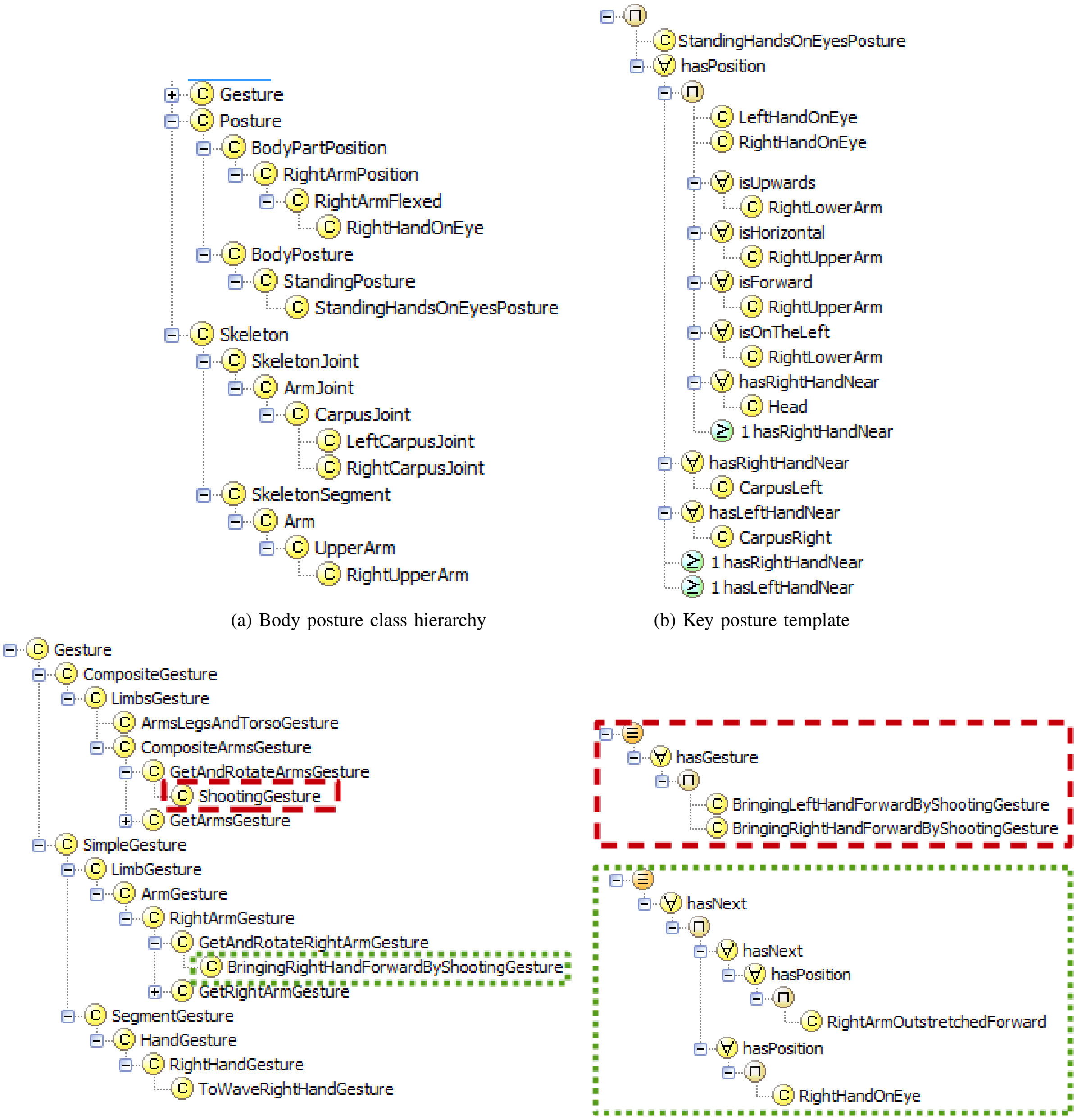
C. Non-standard inferences for semantic matchmaking

The proposed gesture detection approach is based on a deductive inference framework which: (i) exploits machine-understandable annotated descriptions; (ii) allows to infer implicit knowledge from annotations; (iii) adopts the Open World Assumption (OWA, the lack of a feature in a resource representation is not necessarily a constraint of absence). Particularly, matchmaking services are leveraged to: (i) detect key posture best matching retrieved low-level annotation; (ii) derive gesture annotation by applying inference services to detected key posture sequences; (iii) identify the gesture class corresponding to obtained description. The key posture identification is managed as a semantic-based resource discovery on a Knowledge Base, where logic-based inference services provide a similarity ranking. 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 $S_i (i = 1, \dots, n)$ for a given request R, where both request and resources are annotated with respect to a common reference ontology [7]. In the proposed approach, S_i are the key posture templates in the Knowledge Base, 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. On the contrary, *Concept Abduction* (CA) non-standard inference service, originally formalized and applied in e-commerce scenarios [7], is adopted in this work to: (i) provide explanation of outcomes beyond the trivial "yes/no" answer of subsumption tests and (ii) enable a logic-based relevance ranking of a set of available resources for a specific 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. 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. Given S and R in Conjunctive Normal Form, Algorithm 1 finds a minimal solution for CA in $\mathcal{ALN}$ DL in the number of conjuncts in H and computes the corresponding *penalty function* for S with respect to R [7]. In the proposed approach, semantic similarity is computed by normalizing the penalty according to the ontology structure [3] and translating it in a $[0, 100]$ percent ascending scale.

A toy example should clarify the above process. Let us



(a) Body posture class hierarchy

(b) Key posture template

(c) Gesture hierarchy

(d) Gesture definition template

Fig. 3: Posture ontology model

consider the following resources in the Knowledge Base key posture templates .

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.* With reference to the 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 referenced defined classes.

Movement information is inferred by applying CA to pairs of subsequent key postures. For example, assuming G=$S_1 \rightarrow S_2$ is detected, the system automatically infers the following gesture dynamics:

```
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 such that $B \sqsubseteq A$ 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

$\mathbf{H_{S_1,S_2}} \equiv \forall\ hasPosition.(\ \forall\ isHorizontal.(RightLowerArm \sqcap LeftLowerArm) \sqcap \forall\ isDownwards.(RightUpperArm \sqcap LeftUpperArm).$

$\mathbf{H_{S_2,S_1}} \equiv \forall\ hasPosition.(\equiv\ \forall\ isUpwards.(RightLowerArm \sqcap LeftLowerArm) \sqcap \forall\ isHorizontal.(RightUpperArm \sqcap LeftUpperArm).$

H_{S_1,S_2} models the starting horizontal position of arms, which are lifted to vertical position, expressed in H_{S_2,S_1}. Therefore, the overall gesture annotation is retrieved:

$(\mathbf{G})\mathbf{DetectedGesture} \equiv \forall\ hasNext.(\ \forall\ hasPosition.(\ \forall\ isHorizontal.(RightLowerArm \sqcap LeftLowerArm) \sqcap \forall\ isDownwards.(RightUpperArm \sqcap LeftUpperArm)) \sqcap \forall\ hasNext.(\ \forall\ hasPosition.(\equiv\ \forall\ isUpwards.(RightLowerArm \sqcap LeftLowerArm) \sqcap \forall\ isHorizontal.(RightUpperArm \sqcap LeftUpperArm)))).$

Finally, the system can compute semantic similarity between detected movement G and stored gestures (resources) from the repository, computed by the matchmaker solving *Concept Abduction*. Furthermore, the best matching gesture is returned along with the related score and H values.

III. Prototype and evaluation

In order to prove both feasibility and effectiveness of the proposed approach, a software prototype has been implemented, extending the *Kinect Toolbox*[3]. Thanks to the GUI in Figure 4, the tool allows users to compose semantic annotations for body postures and gestures, without requiring specific knowledge of Semantic Web languages and underlying logic-based formalisms. When a subject is facing the Kinect sensor, her/his movements are tracked and skeleton data are retrieved and displayed on the panel (A). Hence, the system allows to process pre-recorded data and by pressing the 'Annotate Posture' button on panel (D) a real-time posture description is shown. Panel (B) provides an intuitive tree-like graphical representation. The reference ontology is loaded and its elements populate the upper part of the panel. Current posture annotation is displayed in the lower portion of panel (B) so that the user can edit it through drag-and-drop of classes and properties from the ontology: context menus appear whenever additional information have to be specified. Finally the description can be saved by clicking on the 'Apply' button just below.

Current body pose is detected as a possible key posture only if it lasts for a tunable *motion sensitivity* parameter. In that case, it is automatically processed for recognition. The embedded lightweight reasoner [2] performs semantic matchmaking between the real-time annotation and each posture instance in the Knowledge Base. As a result, the recognized pose is returned as the one that best matches the Kinect-derived annotation, and it is added to the timeline (C). The sequence of recognized key postures is processed by the embedded reasoner, as explained in Section II-C. Inferred semantic description is displayed in panel (B) while the recognized gesture is added in the list (C), between the key postures composing the sequence.

An experimental campaign was carried out to obtain a preliminary performance evaluation of the proposed method. A subset of gestures was selected from Microsoft Research Cambridge-12 data set [4]. Each gesture was repeated at least 10 times consecutively. Preliminarily, a set of body posture and the associated gestures were defined. Then the prototype 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 similarity threshold of 70%. Results are reported in Table I. For each gesture the F-score with corresponding precision and recall were taken. In general, precision is high (always above 0.8); if posture detection results are good, then also recall is high. The proposed approach was compared to [8], where performance tests on the same gesture set were evaluated against three classifiers: SVM, Naive Bayes and Random Forest. Histogram in Figure 5 shows the data: in three of five cases semantic matchmaking performs better; in two cases it takes the third place. The overall performance is comparable with the more computationally intensive methods SVM and random forest. Higher inter-gesture variance is due to the quality of recognition at the posture level. Unfortunately, a more complete comparison including processing intervals could not be performed, because times were not reported in [8].

IV. Related work

A comprehensive state of the art of activity recognition can be found in [9]. Detection algorithms can be divided into *machine*

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

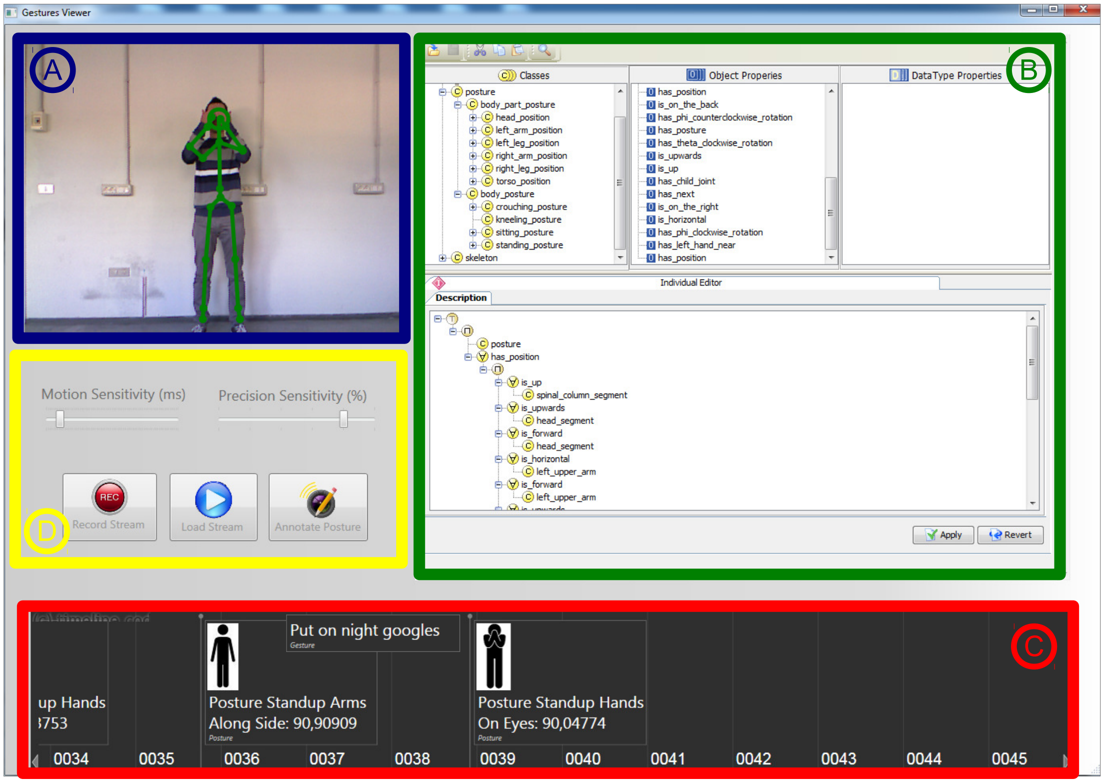


Fig. 4: Prototype tool screenshot

Gesture	F-score	Precision	Recall
Start music/raise volume			
	0,92	0,94	0,90
Crouch or hide			
	0,88	0,95	0,82
Navigate to next menu			
	0,62	0,82	0,50
Take a bow			
	0,66	0,85	0,54
Shoot with a pistol			
	0,81	0,92	0,73

TABLE I: Experimental results

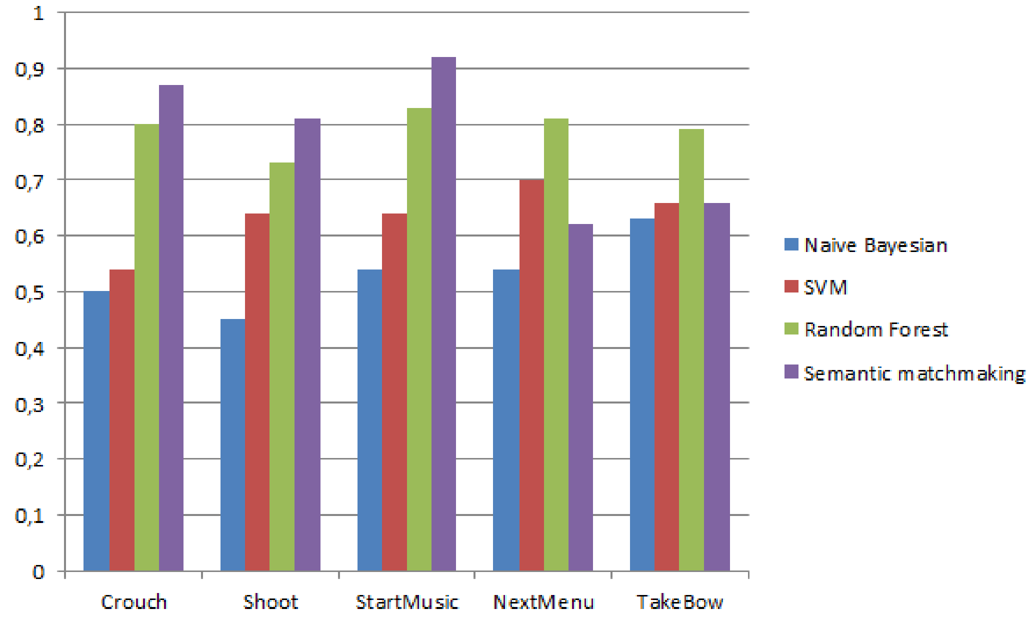


Fig. 5: F-score of four methods

learning and *ontology*-based ones. Methods grounded on machine learning can be either supervised or unsupervised. *Supervised* techniques [5], [10] require a relatively large corpus of labeled data to be built for training, usually by hand. Furthermore, the resulting models achieve good accuracy only for the specific scenarios they are built for. They are not reusable and scalable when individual behavior or environmental conditions change. Therefore, recognition of a large diversity of activities in real-world application scenarios could be deemed as impractical. *Unsupervised* methods [11] try to build 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 availability of large data collections associated with partial human annotation has recently turned the attention to *semi-supervised* learning [12]. By combining small-scale expert labeled data and large-scale unlabeled data based on certain assumptions, semi-supervised learning methods try to find the best tradeoff between system accuracy and required human and computational effort.

Ontology-based activity recognition follows a completely different approach. It exploits knowledge representation for activity and sensor data modeling, and logic-based reasoning to perform activity recognition. Such approaches: (i) use semantically rich formalisms to explicitly define and describe a library of models for all possible instances in a domain; (ii) aggregate and translate sensed data into logical formulae grounded on the above terminology; (iii) perform deductions 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. Ontologies are also a way to share knowledge between researchers of recognition algorithms and application developers, who can expand and adapt the terminology to their particular needs. In order to exploit this benefit, in [13] 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 was proposed in [14], 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 aggregation of elementary concepts through 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. This is why the tool proposed here –even though in a prototype form– lets domain experts expand the core Knowledge Base through a visual workflow. Automated analysis of surveillance video is one of the most frequent applications of ontology-based activity recognition and annotation [14], [15]. Chen and Nugent [9] proposed an ontology-based approach which is more similar to the one described here: Activities of Daily Living (ADL) Description Logics (DL) ontology was produced for activity modeling and reasoning in the context of smart homes. Subsumption was used to enable flexible activity recognition at different levels of detail, depending on the amount of knowledge acquired from the environment. Anyway, as pointed out in [16], standard inference services are not enough in these cases, since recognition/interpretation tasks cannot be considered simply as *classification* tasks, but they are more similar to *model construction* ones. Therefore, in the approach proposed here, subsumption is replaced by the Concept Abduction non-standard DL reasoning task, which actually aims to build a concept (*i.e.*, a model) for missing information whenever a full/subsumption match cannot be achieved.

The main weakness of most logic-based approaches it that they do not offer mechanisms for deciding whether one particular model is more effective than another. In the framework of this paper, recognition is treated as a matchmaking problem returning a score as output, which can be used to compare different ontological models. Besides, the majority of ontology-based proposals have adopted a top-down approach so far, focusing only on high-level activities and events. The approach presented here is complementary, since it allows activity recognition from the bottom up. Hence, it is open to extension toward the upper layers or to integration with other high-level event annotation frameworks.

Several tools aim to support developers of recognition applications, *e.g.*, *KINA* toolkit [17] and *DejaVu* [18]. Conversely, practitioners are the target users of the proposed prototype: the tool helps them build representative postures and gestures for their work domain, by composing visually a high-level description. Furthermore, annotations can be derived from the automatic characterization of the posture and gestures of the user in front of Kinect.

V. Conclusion and future work

The paper introduced a general-purpose framework for semantic-based gesture annotation and recognition. It exploits 3D joint position data provided by a Microsoft Kinect device. A general model was devised to characterize body parts and most common postures, and a corresponding ontology was designed to annotate them formally through Semantic Web languages. The posture and gesture recognition problem have been basically handled as resource discovery grounded on semantic matchmaking, exploiting non-standard inference services. The framework has been implemented in a prototypical tool: early results obtained with respect to a reference dataset provide a promising proof of concept.

Future work aims to enhance the presented framework toward action recognition. This goal will require an extension of both data model and domain ontology, allowing to annotate an action as ordered sequence of gestures. 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 both in terms of accuracy and computational performance.

Acknowledgments

The authors acknowledge partial support of the Italian PON project Puglia Digitale 2.0 and E.T.C.P. Greece-Italy 2007/2013 ARGES (pAssengeRs and loGistics information Exchange System) project.

References

[1] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, “Comparing gesture recognition accuracy using color and depth information,” in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2011, p. 20.

[2] M. Ruta, F. Scioscia, E. Di Sciascio, F. Gramegna, and G. Loseto, “Mini-ME: the Mini Matchmaking Engine,” in *OWL Reasoner Evaluation Workshop (ORE 2012)*, ser. CEUR Workshop Proceedings, I. Horrocks, M. Yatskevich, and E. Jimenez-Ruiz, Eds., vol. 858. CEUR-WS, 2012, pp. 52–63.

[3] M. Ruta, E. Di Sciascio, and F. Scioscia, “Concept Abduction and Contraction in Semantic-based P2P Environments,” *Web Intelligence and Agent Systems*, vol. 9, no. 3, pp. 179–207, 2011.

[4] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, “Instructing people for training gestural interactive systems,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 1737–1746. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208303

[5] M. Raptis, D. Kirovski, and H. Hoppe, “Real-time classification of dance gestures from skeleton animation,” in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA ’11. New York, NY, USA: ACM, 2011, pp. 147–156. [Online]. Available: http://doi.acm.org/10.1145/2019406.2019426

[6] J. Renz and D. Mitra, “Qualitative direction calculi with arbitrary granularity,” in *In Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*. Springer, 2004, pp. 65–74.

[7] S. Colucci, T. Di Noia, A. Pinto, A. Ragone, M. Ruta, and E. Tinelli, “A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces,” *International Journal of Electronic Commerce*, vol. 12, no. 2, pp. 127–154, 2007.

[8] H. Zhang, W. Du, and H. Li, “Kinect gesture recognition for interactive system.”

[9] L. Chen and C. Nugent, “Ontology-based activity recognition in intelligent pervasive environments,” *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009. [Online]. Available: http://dx.doi.org/10.1108/17440080911006199

[10] K. Biswas and S. K. Basu, “Gesture Recognition using Microsoft Kinect®,” in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*. IEEE, 2011, pp. 100–103.

[11] T. Hunh and B. Schiele, “Unsupervised discovery of structure in activity data using multiple eigenspaces,” in *Location- and Context-Awareness*, ser. Lecture Notes in Computer Science, M. Hazas, J. Krumm, and T. Strang, Eds. Springer Berlin Heidelberg, 2006, vol. 3987, pp. 151–167. [Online]. Available: http://dx.doi.org/10.1007/11752967_11

[12] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu, “A Generic Framework for Video Annotation via Semi-Supervised Learning,” *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1206–1219, 2012.

[13] S. Saad, D. De Beul, S. Mahmoudi, and P. Manneback, “An Ontology for video human movement representation based on Benesh notation,” in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*. IEEE, 2012, pp. 77–82.

[14] A. François, R. Nevatia, J. Hobbs, R. Bolles, and J. Smith, “VERL: an ontology framework for representing and annotating video events,” *MultiMedia, IEEE*, vol. 12, no. 4, pp. 76–86, Oct.-Dec.

[15] J. C. SanMiguel, J. M. Martinez, and A. Garcia, “An ontology for event detection and its application in surveillance video,” in *Advanced Video and Signal Based Surveillance, 2009. AVSS’09. Sixth IEEE International Conference on*. IEEE, 2009, pp. 220–225.

[16] J. Gómez-Romero, M. A. Patricio, J. García, and J. M. Molina, “Ontology-based context representation and reasoning for object tracking and scene interpretation in video,” *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7494–7510, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2010.12.118

[17] B. Reis, J. a. M. Teixeira, F. Breyer, L. A. Vasconcelos, A. Cavalcanti, A. Ferreira, and J. Kelner, “Increasing kinect application development productivity by an enhanced hardware abstraction,” in *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS ’12. New York, NY, USA: ACM, 2012, pp. 5–14.

[18] J. Kato, S. McDirmid, and X. Cao, “Dejavu: Integrated support for developing interactive camera-based programs,” in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’12. New York, NY, USA: ACM, 2012, pp. 189–196.