Journal of Medical Robotics Research, Vol. 2, No. 3 (2017) 1740007 (14 pages)
© World Scientific Publishing Company
DOI: 10.1142/S2424905X17400074

Journal of Medical Robotics Research
http://www.worldscientific.com/worldscinet/jmrr

Journal of Medical Robotics Research

Toward a Knowledge-Driven Context-Aware System for Surgical Assistance

Hirenkumar Nakawala*, Giancarlo Ferrigno, Elena De Momi

Department of Electronics, Information and Bioengineering (DEIB)
Politecnico di Milano, Piazza Leonardo da Vinci, Milan 20133, Italy

Complex surgeries complications are increasing, thus making an efficient surgical assistance is a real need. In this work, an ontology-based context-aware system was developed for surgical training/assistance during Thoracentesis by using image processing and semantic technologies. We evaluated the Thoracentesis ontology and implemented a paradigmatic test scenario to check the efficacy of the system by recognizing contextual information, e.g. the presence of surgical instruments on the table. The framework was able to retrieve contextual information about current surgical activity along with information on the need or presence of a surgical instrument.

Keywords: Context-awareness; ontology; robotic surgery; object recognition; surgical assistance system.

1. Introduction

Surgical procedures are becoming more complex with the advent of new technologies, such as actuators, e.g. robots, helping performing part of the intervention. There is also an increasing availability of intra-operative sensors, e.g. optical trackers, endoscopic cameras working during the surgery. Intra-operative sensors increase the information overload, which may outpace surgeon's cognitive capacity to analyze the information to effectively follow the course of the intervention when required to make a surgical decision. There are different methods to represent conceptual knowledge, such as first-order logic, production rules, semantic nets, frame systems, Bayesian network, and ontologies [1]. Surgical interventions and surgical processes were, in the past, represented by using ontology [2, 4, 5, 18]. In order to be processed by computers, ontologies can be marshaled into known data serialization formats, such as eXtensible Markup Language (XML). Medical ontologies, such as Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) [3] are already being used in health information systems to make it consistent, transferrable, and interoperable by providing a common set of concepts. However, surprisingly, ontology has not yet been widely implemented in the domain of context-aware surgery due to the complexity of surgical processes and lack of standardization for ontologies in surgery domain. We believe that ontology-based knowledge formalism has an utmost importance to reduce surgical complications by providing contextual information at a specific time, e.g. information about a surgical instrument in use based on the step currently in progress. The major role of ontology is to formalize and structure the domain specific knowledge e.g. surgery, to retrieve and integrate data from sensors, e.g. for image processing, and to facilitate interoperability. Ontology-based knowledge could help intelligent systems to understand the interventions, using surgical process models, and the operating scenario, using image processing tools. Ontologies can be used to train and to work along with surgeons for aiding clinical decision making through novel human–computer interfaces. In this paper, we aim at creating a context-aware system, which uses ontology as a knowledge base

Received 29 February 2016; Revised 22 June 2016; Accepted 2 October 2016; Published xx xx xx. Published in JMRR Special Issue on CRAS 2015. Guest Editors: Leonardo Mattos, Paolo Fiorini, Emmanuel Benjamin Vander Poorten and Benoît Herman.

Email Address: **hirenkumar.nakawala@polimi.it*

NOTICE: Prior to using any material contained in this paper, the users are advised to consult with the individual paper author(s) regarding the material contained in this paper, including but not limited to, their specific design(s) and recommendation(s).

for the recognition of surgical instruments and phases at a specific surgical activity as per surgeon's requirement during the surgical procedure.

Different methods have been explored to create contextual awareness during surgical interventions. Katić *et al.* [4], as an example, used Description Logic (DL) to achieve knowledge formalism for laparoscopic procedures, such as cholecystectomy, and rule-based intraoperative context-awareness to recognize surgical phases. Katić *et al.* [5] also developed the LapOntoSPM ontology for laparoscopic surgeries, which was sharable, extensible, and inter-operational with established knowledge representations. LapOntoSPM ontology was used for the phase recognition to help situation interpretation. The ontology in the rule-based system was evaluated with recorded surgical videos. Agrawal *et al.* [6] developed a context-aware fuzzy rule-based perioperative system to infer surgical events, such as the onset of anesthesia, from Radio-Frequency Identification (RFID) tags and from monitoring equipment such as pulse oximeter. Neumuth and Meißner [7] used redundant, complementary, and co-operative information fusion methodology to detect, identify, and localize surgical instruments in the interventional suite. The performance of information fusion-based system was good despite the interference with metal surgical instruments on RFIDs. Bardram *et al.* [8] used a similar sensor platform, consisting RFID and other physiological sensors in the operating suite, and machine learning approach for activity recognition, e.g. instruments in use during the surgical intervention. However, the sensor platform still had issues with reflection and shielding with RFID technology. Kassahun *et al.* [9] developed a classifier on ontology-based and genetic-based machine learning to classify epilepsy types and their localization using ictal symptoms. Other researchers also used machine learning such as Hidden Markov Models (HMM) [10], statistical modeling [11], and signal processing [12] of laparoscopic videos to extract information and to recognize contexts during surgical procedures. De Momi *et al.* [13] developed an autonomous path planner by providing a fuzzy patient risk description, through multi-model information and surgeon-dependent rules stored in the knowledge base, for brain surgery. Most of these systems did not explicitly use the knowledge but processed the data and information only. In addition, most of the previous approaches, which have been used for the detection of surgical phases and related information through ontology, miss the retrieval of specific contextual information of the procedure that can arise from the integration of ontology, data processed through 3D image processing, and semantic technologies.

Our work is focused on modeling of a percutaneous procedure, the "Thoracentesis". Pleural effusion is a debilitating condition, often associated with other diseases, in which there is a build-up of fluids between lung tissue and pleural space. Common symptoms of pleural effusion can include pleuritic chest pain, coughing, and dyspnea [14]. Removal of fluid from the pleural cavity is performed by an invasive procedure in which a needle is inserted into the chest cavity and the fluid withdrawn using a syringe [14]. Procedure-related complications are a major problem and can range from pain, dry cough, no fluid return or subcutaneous collection which affects at most 33% people, to life-threatening complications such as pneumothorax, pulmonary edema, unintentional puncture of spleen or liver and sheared off catheter in the pleural space, and in some exceptional conditions winging of the scapula [15]. Although Thoracentesis is a very simple procedure, procedure-related complications are higher than expected.

The aim of our work is to create a reliable and repeatable aid, such as context-aware system, to assist the novice surgeons training in performing needle Thoracentesis. The importance aspects of needle Thoracentesis are the insertion of a 50 mL syringe needle for withdrawing the pleuritic fluid without injuring internal organs and the identification of an accurate anatomical landmark on the chest. For performing needle Thoracentesis, the surgeon needs technical skills, such as dexterity, procedural knowledge, as well as nontechnical skills, such as cognitive skills (context awareness, decision-making and planning) [16, 17]. Mental models, surgeon's knowledge, and the experience, stored in the memory, influence judgments related to intraoperative context awareness. Our system provides cognitive support during the surgical training to novice surgeons in Thoracentesis performance. Therefore, considered as an aid to cognitive decision making, the system will eventually reduce medical errors, iatrogenic complications and overall improve the patient care. The paper also demonstrates the use of open source technologies to create a context-aware system for surgical assistance.

2. Methods

In our scenario, surgeons query the procedure ontology (Sec. 2.1), which includes information about all the instruments required during Thoracentesis. Based on the queried result, the developed application (Sec. 2.3) represents the sequence of steps and activities during the procedure and recognizes (Sec. 2.2) the instrument, e.g. a syringe or a surgical-swab on the surgical stand, in use during a specific surgical step. The framework automatically identifies this information. The system further suggests the next step and the next instrument(s) to be used based on the semantic relations, e.g. "next step" and "instrument in step" grounded in the ontology.

Our framework [18] integrates ontology-based surgical knowledge (Sec. 2.1), data acquired through 3D image

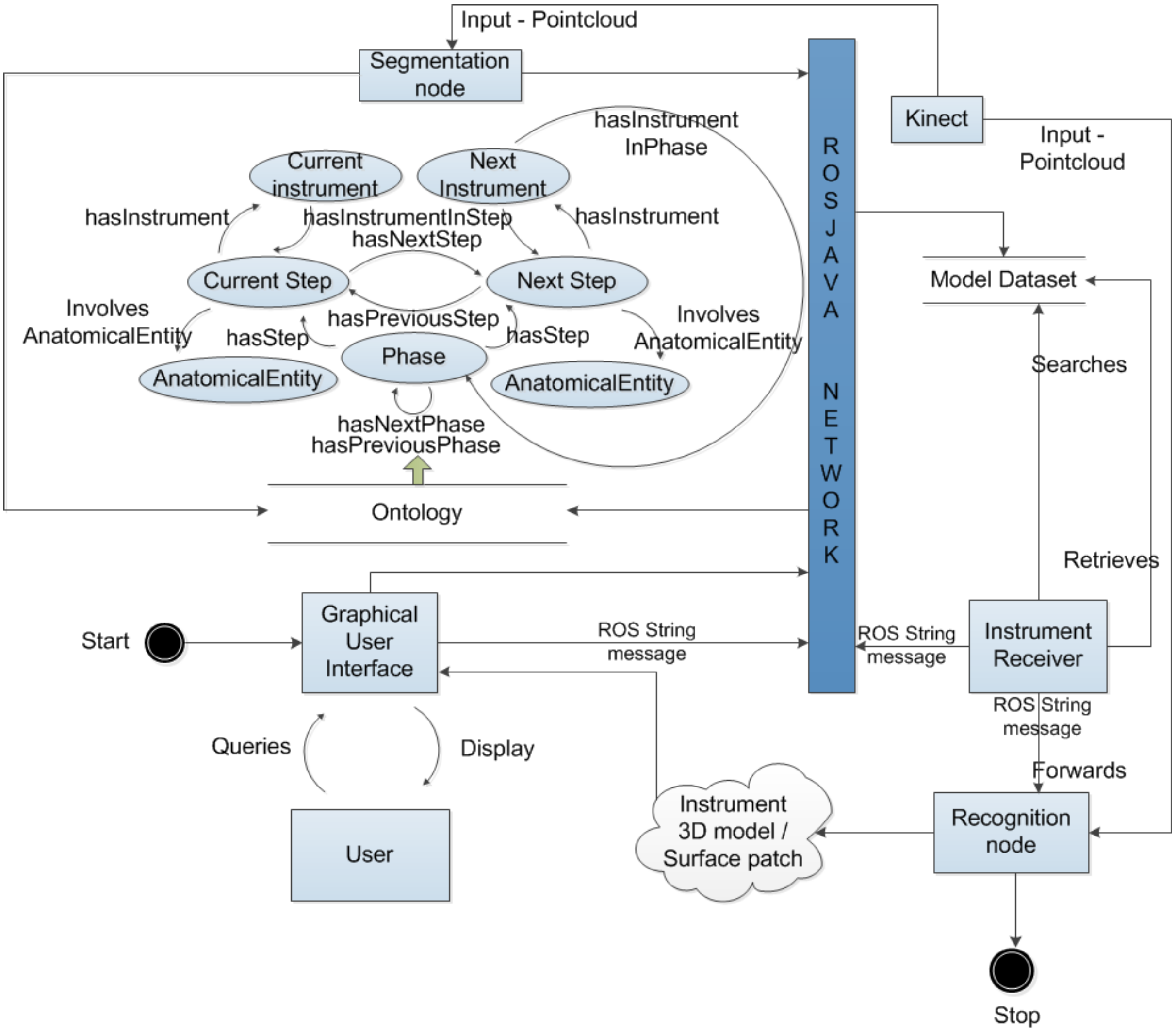


Fig. 1. Data-flow diagram: The data-flow underlying the scenario mentioned above is shown. At first, user queries the surgical step through the Graphical User Interface (GUI). The queried step is forwarded to the component containing ontology through a ROS message. After reasoning, ontology component finds an instrument instance corresponding to the step based on semantic relations. After identifying the instrument instance, the ontology component publishes a message containing the information on instrument instance. After that, "Instrument Receiver" finds the point-cloud surface patch models in the dataset based on instrument instance, which has similar nomenclature as the instrument instance grounded in ontology. Further to that, "Instrument Receiver" forwards the obtained information to the "recognition node". The "recognition node", which processes template-matching algorithm, then finds an instrument on the table and represents the surgical phase and instrument information on the GUI. In our framework, "segmentation node" segments instrument surface patches from 3D point-cloud and saves them in datasets to be used by "recognition node".

acquisition and processing (Sec. 2.2), and a user query interface (Sec. 2.3) to create a context-aware system. As shown in Fig. 1, the overall system framework was realized through ROSJAVA package [19], which is an open source project in pure java for integration with Robot Operating System (ROS) [19]. It allows java applications to interface with ROS nodes, topics, messages, services, and parameters. Our framework uses ROS messages, to pass information among the "ontology", which processes procedural knowledge and surgical activities, the "segmentation node" which processes segmentation algorithms, and the "recognition node" which processes recognition algorithms. "Instrument Receiver" is an intermediary node that manages messages exchanges between framework components.

2.1. *Surgical procedure*

Knowledge representation is an imperfect approximation [1] of real-world entities. In our case, generalized concepts for Thoracentesis were tailored to represent contextual information by integrating entities of the procedure and sensory data from an imaging sensor.

Ontology for Thoracentesis was constructed using a top-down approach, where most general concepts of the domain, such as phases (e.g. "Anesthesia") were analyzed first and thereafter specialized concepts, such as actions (e.g. "WithDrawLargeSyringe"), were implemented. The needed information about Thoracentesis was obtained from a journal paper [20], online web resources, which were verified through HONCode [21] for health information authenticity and analyzed using the methodology described in [2], and an opinion from a physician. Ontology encompassed surgical entities, for example surgical steps, surgical process model, and information about different spatiotemporal events that occurs during Thoracentesis procedure. After identifying appropriate classes, the procedure was formalized using an approach similar to [22], where logical sentences were divided into triplets in the format of "Phase (Instrument, Step, Body Structure)", specified for each surgical phase. For example, withdrawal of a large syringe (50 mL) from the area of insertion on intercostal space is expressed as "Closure (LargeSyringe, WithdrawLargeSyringe, AreaOfInsertionIntercostal)". "Closure" represents the phase of Thoracentesis. It involves the surgical activities carried out during the end of an intervention, where the surgeon removes the syringe from chest cavity after all the fluid is withdrawn. All of these classes were considered as sub-classes of the procedure class. Actions were considered as temporal parts of the phases. The instances of the instrument's models were also created in the ontology (e.g. "LargeSyringe.pcd") and configured in the application developed for detection. We also added seven data properties to define variables for the instruments point-cloud model "data" (range:double), "fields" (range: string), "height" (range:integer), "points" (range:integer), "size" (range:integer), "type" (range:string), "width" (range:integer). Instrument's model centroid information ("x", "y", and "z") is also specified. Instrument's instances have been assigned this data property. All the procedure-related classes, object properties, and data properties were specified in Terminology Box (TBOX), while real-world procedure events were specified in the Assertion Box (ABOX) using indicative propositions. Each of these latter represents knowledge for the instances of the surgical entities. Automated description logic-based reasoners constantly check the ontology. We also used Jena's generic reasoner to reason over the ontology and performed manipulation over the ABOX in the application developed for the evaluation. Ontology provides a knowledge base and allows the surgeon to query on phases, steps, body structures, and instruments. Body structures or anatomical entities, related to Thoracentesis, are selected by analyzing the procedure as described in the literature and by suggestions of an expert. The relevant subset of classes, pertinent to Thoracentesis, are extracted from SNOMED-CT and merged with the developed ontology using Protégé [23]. The use of SNOMED-CT ensures inter-operability with other health information systems. Use of SNOMED-CT also allows making a balance between vocabulary reusability and granularity required for domain knowledge. The ontology was serialized using (Resource Development Framework) RDF/XML format by Protégé. RDF/XML allows ontology to be processed by a wide range of XML and RDF processing tools and thus facilitates easy integration with other semantic web technology platforms.

2.1.1. *Surgical workflow*

The surgical workflow consists here of sequence of all the activities performed during Thoracentesis procedure and their connection to the higher levels of granularity, e.g. surgical phases. Our current approach represents individual occurrences during the intra-operative procedure and hence, the ontology is described with a limited number of instances. The workflow does not contain people, their roles or any other perioperative and administrative tasks in execution of surgery. The workflow only represents hierarchical steps of the activities performed in the procedure, thus, it keeps the computational time short, by excluding surgical administrative information, which is beneficial for the quick ontology reasoning by the reasoners. As shown in Fig. 1, steps are represented by an instance of the surgical activity, which are connected to the instrument through "hasInstrumentInStep" relation and to the body structure through "involvesAnatomicalPart". Moreover, each activity is associated with the phases through "isInPhase" relation. Each surgical phase is also connected to the instrument through "hasInstrumentInPhase" and steps through "hasStep" relation. To form a surgical plan, which are the possible sets of surgical activities divided into phases, different classes and its instances are related to each other by object properties, which help to identify temporal order of the phases and steps. These properties are specified in chronological orders such as "hasNextStep", "hasPreviousStep", "hasNextPhase", "hasPreviousPhase". Surgical plan enforces constraints for possible temporal sequences of the surgical procedure and allows consistent guidance to surgeons during the surgical intervention.

2.2. ***3D image processing***

2.2.1. *Acquisition system and system architecture*

A Microsoft Kinect for Xbox 360 (Microsoft, Redmond, WA, USA) was used to acquire the registered point-cloud from the scene, which includes the tabletop, several surgical instruments, and materials. We used the Open Natural Interaction library (OpenNI) [24] to transform raw data from Kinect into point clouds for processing through algorithms included in the Point Cloud Library (PCL) [25] in ROS.

2.2.2. *Segmentation algorithms*

The segmentation process is described in Fig. 2. After the acquisition of registered point-cloud, the user chooses the segmentation algorithms (either RANSAC [26] based plane segmentation [27] or color-based region growing segmentation [28]) based on object's features e.g. object's shape.

Conversely, a set of instrument models is also obtained by using RANSAC-based plane segmentation from the 3D meshes reconstructed from the object reconstruction algorithm, e.g. "KinectFusion" [29], to obtain better instrument's surface patches. Eventually, user saves the identified models in database.

(i) RANSAC-based plane segmentation

To downsample the point-clouds and to approximate the region of interest, voxel-grid and pass-through filters were implemented to decrease point-cloud density and to remove the outlier points such as walls [30] respectively. We used the RANSAC algorithm [26, 27], which is a widely used computer vision technique to segment planes from data points, even in case of outliers. Moreover, after segmentation, remaining outliers were removed with the statistical outlier removal method [31].

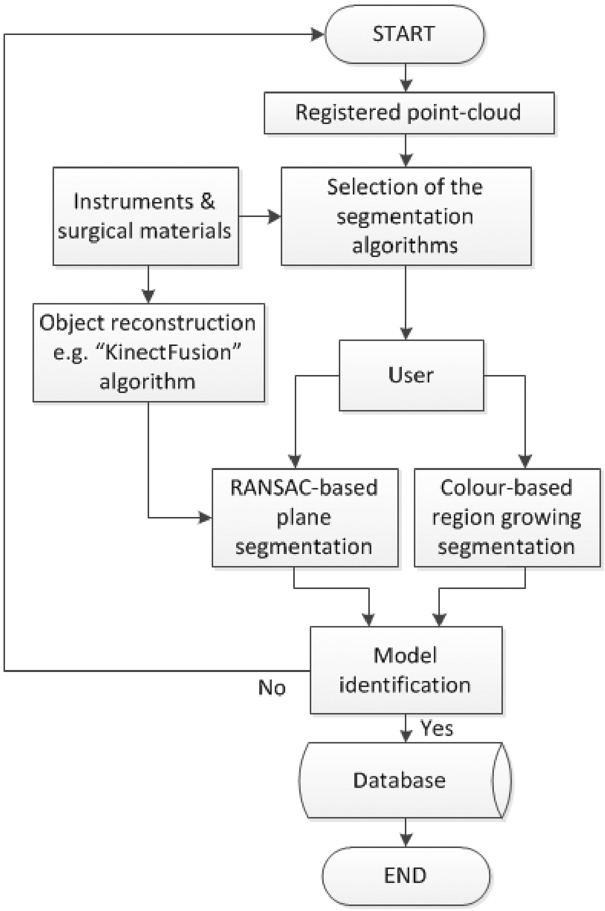

START
Registered point-cloud
Instruments & surgical materials
Selection of the segmentation algorithms
Object reconstruction e.g. "KinectFusion" algorithm
User
RANSAC-based plane segmentation
Colour-based region growing segmentation
Model identification
No
Yes
Database
END

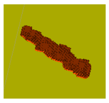
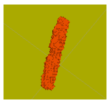
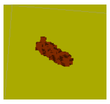
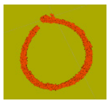
Fig. 2. Flowchart representing the segmentation process of 3D object surface patches.

(ii) Color-based region growing segmentation

We also used color-based region growing segmentation [28], which groups together points that have similar color characteristics. There are two post-processing steps in this algorithm. In the first step, neighboring clusters were found out using point distance threshold, then a point color threshold was defined which was used for testing the points colors. A value for cluster color threshold was also defined through ROS dynamic reconfigure package which was useful during the cluster merging process and in the second step, clusters, whose size was smaller than minimum size specified, were merged with their neighbors.

After the segmentation, we selected instrument's surface patches for experiments and saved them in the file-system database. Examples of such surface patches are shown in Table 1. We used the state-of-the-art "KinectFusion" algorithm [29] to reconstruct dense surface meshes of the objects lying on the table. Then, we converted the surface mesh into a point-cloud data format to process by PCL. After segmenting the point-cloud, we extracted partial object models of the instruments through the Euclidean cluster extraction method [32] with RANSAC-based plane segmentation. A cluster tolerance was set to divide segmented point-cloud in the separate object clusters. KdTree [33] was used to search through the entire point-cloud for cluster extractions. We

Table 1. Instruments and their point-clouds.

Instrument	Model	Dimension with points information
A syringe (50 mL)		Approx. 17.0 cm × 5.0 cm × 3.5 cm Points: 853
Needles		Approx. 8.0 cm (length); 14G, 16G, 18G (gauge) Points: 548
A surgical swab		Approx. 10.0 cm × 10.0 cm × 1.0 cm Points: 8742
A small syringe (10 mL)		Approx. 10.0 cm × 2.5 cm × 2.0 cm Points: 405
A flexible catheter		Approx. 40.0 cm (length) Points: 2138

verified each of the segmented surface patches by visual inspection in ROS Rviz module and then saved them in the file system database.

2.2.3. *Template matching for instrument recognition*

Object recognition is the process of identifying a model of the object in the environment. 3D object recognition methods can be: (i) appearance-based methods, where objects matched against templates or exemplars of the objects to perform recognition, e.g. edge matching, and (ii) feature-based methods, where algorithms find an object in the environment by matching its image features with object features such as pose consistency. We used SAmple Consensus Initial Alignment (SAC-IA) [34], which tries to maintain the same geometrical relationship of the correspondences candidates, selected based on pairwise distances between clouds, from a scene point-cloud dataset, hereafter referred to as a scene, to align an object template [34]. The error metric for the rigid transformation is determined using Huber penalty measure [34]. We used Fast Point Feature Histogram (FPFH) [34], as feature descriptor, to describe local geometry around the points. The FPFH considers the direct connections between the corresponding candidates and its neighbors. The latter is pose invariant and its discriminative power makes it a good candidate for point correspondence search in 3D registration.

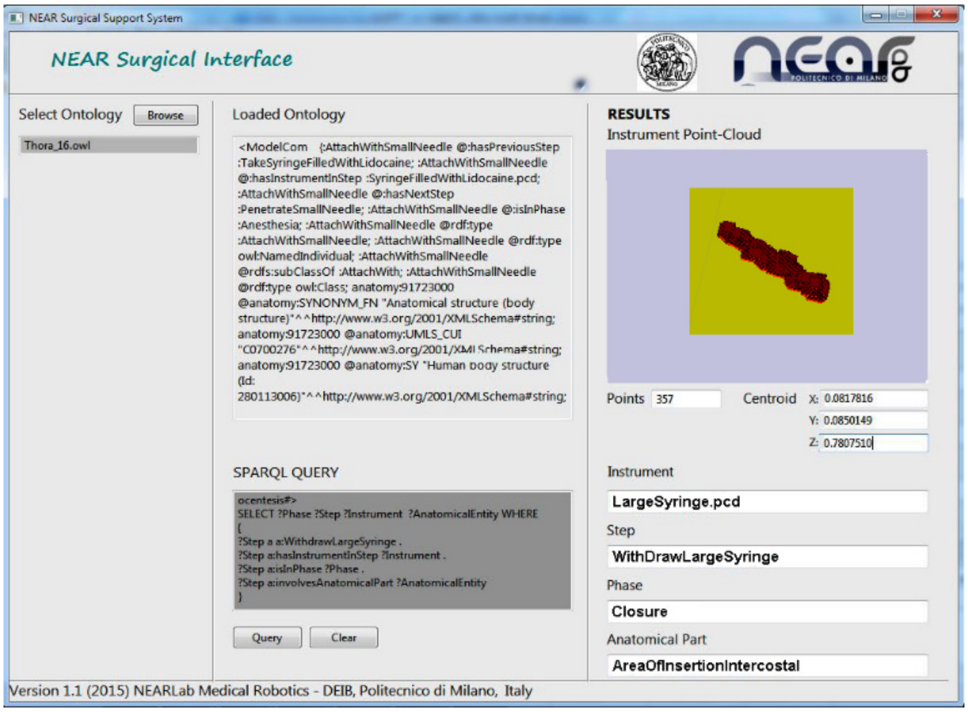
The surgical workflow retrieves instrument's models, stored in the local databases as templates, based on clinician's requirements at a specific instance of phase and step of the procedure. After obtaining the results, the model's information is forwarded to the template matching node implemented in the ROS, which executes the above-mentioned algorithm for aligning the instrument's model and the real instrument present in the scene. The instrument identification provides the context to users for identifying the instruments on the table, which will be helpful to novice surgeons. Moreover, the detected instrument can also be represented as the 3D model to guide the surgeons further on the instruments along with the information on the surgical phase.

2.3. *User application interface - NEAR surgical interface*

2.3.1. *Graphical user interface*

The developed Graphical User Interface (GUI), shown in Fig. 3, helps to reason and query the ontology. The application contains a section for ontology, query (SPARQL language [35]) and results (Instrument point-cloud information such as points and model's centroid information, surgical phases, steps and body structures/anatomical entity).

The user can load the ontology or the ontology can be pre-loaded before the start of the procedure. The



Select Ontology: Thora_16.owl

Loaded Ontology

```
<ModelCom   {:AttachWithSmallNeedle @:hasPreviousStep
:TakeSyringeFilledWithLidocaine; :AttachWithSmallNeedle
@:hasInstrumentInStep :SyringeFilledWithLidocaine.pcd;
:AttachWithSmallNeedle @:hasNextStep
:PenetrateSmallNeedle; :AttachWithSmallNeedle @:isInPhase
:Anesthesia; :AttachWithSmallNeedle @rdf:type
:AttachWithSmallNeedle; :AttachWithSmallNeedle @rdf:type
owl:NamedIndividual; :AttachWithSmallNeedle
@rdfs:subClassOf :AttachWith; :AttachWithSmallNeedle
@rdf:type owl:Class; anatomy:91723000
@anatomy:SYNONYM_FN "Anatomical structure (body
structure)"^^http://www.w3.org/2001/XMLSchema#string;
anatomy:91723000 @anatomy:UMLS_CUI
"C0700276"^^http://www.w3.org/2001/XMLSchema#string;
anatomy:91723000 @anatomy:SY "Human body structure
(Id:
280113006)"^^http://www.w3.org/2001/XMLSchema#string;
```

SPARQL QUERY

```
ocentesis#>
SELECT ?Phase ?Step ?Instrument  ?AnatomicalEntity WHERE
{
?Step a a:WithdrawLargeSyringe .
?Step a:hasInstrumentInStep ?Instrument .
?Step a:isInPhase ?Phase .
?Step a:involvesAnatomicalPart ?AnatomicalEntity
}
```

RESULTS

Instrument Point-Cloud

Points: 357

Centroid X: 0.0817816

Y: 0.0850149

Z: 0.7807510

Instrument: LargeSyringe.pcd

Step: WithDrawLargeSyringe

Phase: Closure

Anatomical Part: AreaOfInsertionIntercostal

Version 1.1 (2015) NEARLab Medical Robotics - DEIB, Politecnico di Milano, Italy

Fig. 3. NEAR Surgical Interface. The loaded ontology, a SPARQL query and identified instrument, with information about cloud-points and its centroid, is linked to instrument's instances that can be forwarded for the recognition.

software was configured, with local resources, through a key-value pair method [36], e.g. to obtain template model from the database.

The highly configurable software allows manipulating the ontology for different surgical procedures if ontology is constructed using the triplets of phases as described before. The software specifically searches resources in the ontology graph for phases, steps, body structures, and instruments. The user interface is also helpful to identify user-specific requirements of instruments. After the application retrieves the information about the instrument, its model information is forwarded to the recognition algorithm to find out an instance of the instrument in the scene. A surgeon can query context about a single instrument at the specific step (e.g. "Anesthesia" phase has more than one action and instrument, however system will allow querying at the specific instance, e.g. "step:InjectWithLargeSyringe", and retrieve the information about 10 mL syringe). Figure 4 represents the information-flow between the system components.

2.3.2. *Query*

We implemented the queries to retrieve information about the instruments used in specific steps and surgical phases while acting on a specific body structure. For example, information about instrument can be extracted using the query construct as shown in Fig. 5.

Most of the SPARQL queries contain a set of triplets, which are required to query the RDF graph data structure, i.e. the ontology, which we recall, consists of triplets of phases. SPARQL query matches the similar patterns between RDF sub-graph and requests triplets, which retrieve the queried information.

We used Jena API [37] along with GUI provided by Java Foundation Classes (JFC)/Swing [38] to query, reason and update the ontology. Figure 6 summarizes the framework components and corresponding open-source technologies used.

2.4. *Experimental protocol*

In order to verify the system performances, we implemented a set of experiments using the setup shown in Fig. 7. The purpose of our experiments was to check usability of the ontology as a knowledge base to retrieve surgical information and to find contextual information related to specific surgical activities detected using image processing algorithms. Our experimental protocol was divided into three steps, ontologies, segmentation algorithms and recognition algorithm, which are explained in the next subsections.

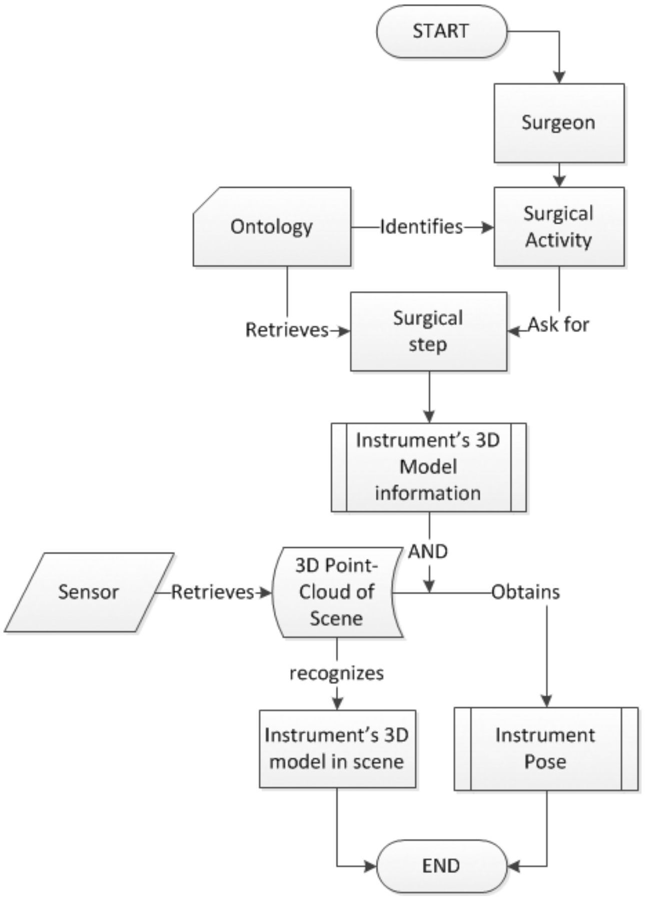
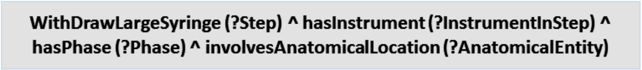


Fig. 4. The flowchart represents the information-flow in our software framework, where surgeon asks about information on a step currently in progress to the ontology, which retrieves information on instrument's models that may be in use. After retrieving the instrument model, the system retrieves the current scene of the surgical table on which the system recognizes the instrument and obtains its pose information.

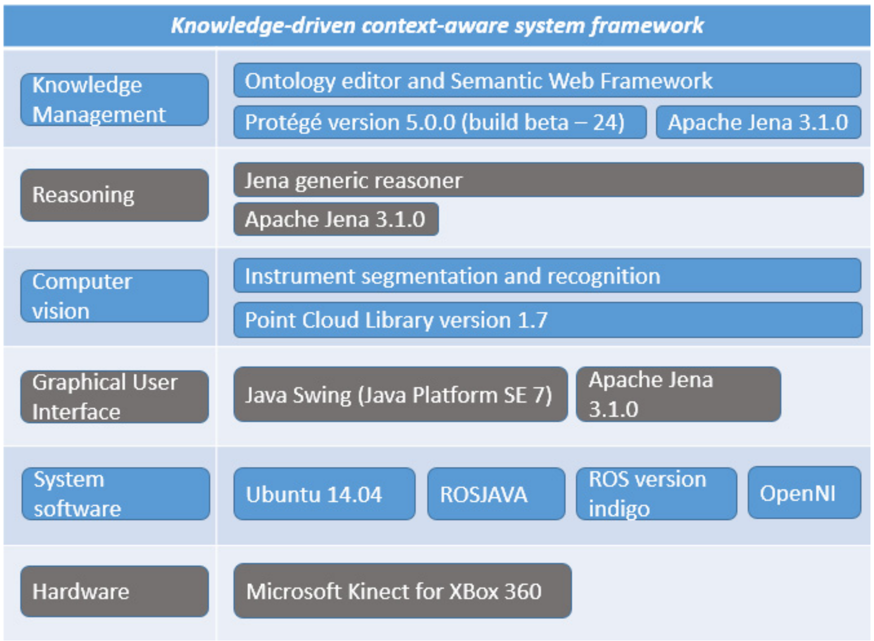
(i) The ontology for Thoracentesis is uploaded in the application as shown in Fig. 3. The user queries the ontology through SPARQL after the ontology is loaded in the interface. As an example, as shown in Fig. 3, user queries the system to find information about an instrument and its point-cloud, lying on the table (e.g. 50 mL syringe), which is withdrawn from the chest during the "Closure" phase at the procedure step "WithDrawLargeSyringe" where the step is decided by the surgeon. The GUI also shows information about current step and involved body structure to provide assistance during surgery.

The segmented instrument model's template is selected based on the instances created in the ontology (e.g. "LargeSyringe.pcd"). To evaluate the



WithDrawLargeSyringe (?Step) ^ hasInstrument (?InstrumentInStep) ^ hasPhase (?Phase) ^ involvesAnatomicalLocation (?AnatomicalEntity)

Fig. 5. Query for identification of an instrument 50 mL syringe.



Knowledge-driven context-aware system framework

Knowledge Management	Ontology editor and Semantic Web Framework Protégé version 5.0.0 (build beta – 24) Apache Jena 3.1.0
Reasoning	Jena generic reasoner Apache Jena 3.1.0
Computer vision	Instrument segmentation and recognition Point Cloud Library version 1.7
Graphical User Interface	Java Swing (Java Platform SE 7) Apache Jena 3.1.0
System software	Ubuntu 14.04 ROSJAVA ROS version indigo OpenNI
Hardware	Microsoft Kinect for XBox 360

Fig. 6. Software stack.

ontology, we iteratively checked it through "FaCT++" [39] and "Pellet" [40] reasoners, which are in-built plug-in in Protégé, during the building process to minimize the inconsistencies in the ontology and through different queries during building and after inclusion of all assertions. If the ontology was found to be inconsistent, then it was incrementally updated or adjusted. We computed True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN), by asking positive and negative queries, by comparing surgical steps and activities of Thoracentesis procedure described in the literature, as a gold-standard, to calculate the error metrics.

We computed the classification sensitivity, which represents the proportions of positives retrievals of information about the instruments that are associated with steps and phases. Specificity represents the proportions of the negative retrievals of information about the instruments that are not associated with step and phase. Accuracy represents the proportion of true results (TP and TN) among the total numbers of the query performed.

(ii) To verify the segmentation algorithms accuracy, we segmented instrument clusters in different poses as mentioned in Table 1. We also used different environmental conditions, such as in the natural sun light, hereafter referred as "without illumination", and with white fluorescent illumination to check the repeatability of the segmentation algorithms on raw point-clouds. It has been proven that "KinectFusion" is suitable for fast acquisition of medium scale scenes to create ground-truth images [41]. Further to test the plane segmentation algorithm, we acquired 10 surface meshes, where the scene contains an instrument, e.g. surgical swab, in different poses on the surgical stand/table-top. We converted meshes into point clouds to be processed by the plane segmentation algorithm.

To create the ground truth, we manually segmented point-clouds by extracting the points that fall inside of a 2D polyline once it is projected on screen. Then we segmented the identical converted point cloud with the help of plane segmentation algorithm. After obtaining surface model patches of the instrument by the latter from scene, we computed ~~approximate~~ Euclidean distance between manually segmented model's point cloud, as a reference, to a target model's point cloud obtained using plane segmentation from scene. The distance is computed by the shortest nearest point distance between both the segmented models to assess accuracy of the plane segmentation algorithm, where nearest neighbor point Euclidean distance D is shown in Eq. (1).

$$D(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}. \tag{1}$$

In Eq. (1), k represents the number of neighbors, and x_i and y_i are the source and target points. We fixed a default set of parameters, as mentioned in Table 2,

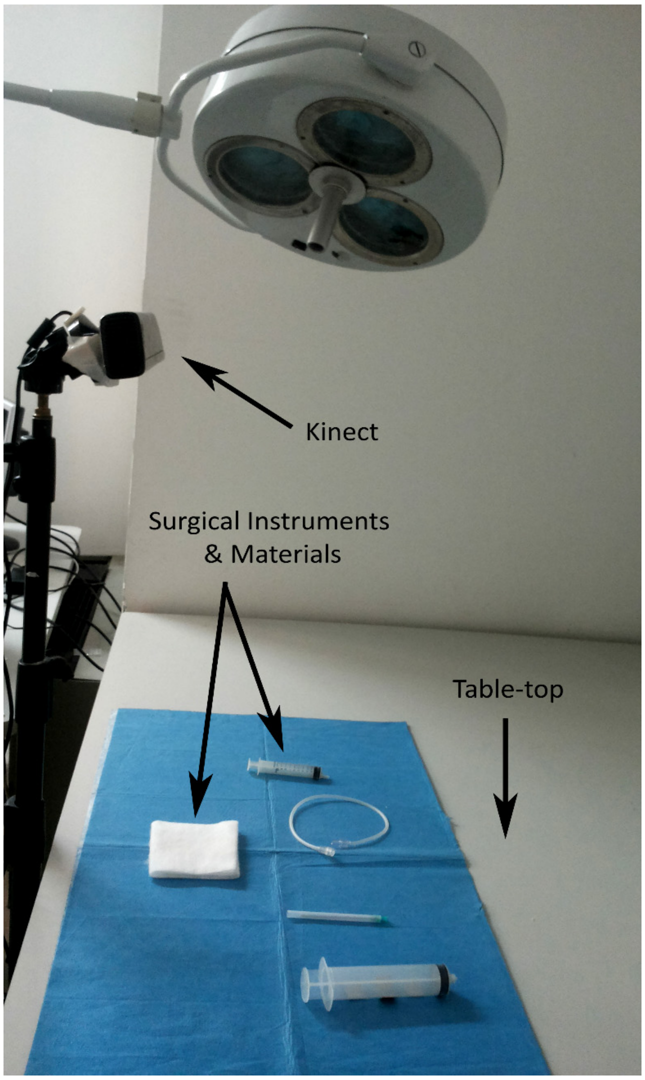


Fig. 7. The experimental setup includes a Microsoft Kinect, the table-top and surgical instruments. Kinect was setup approximately 60 cm away from the table-top. The figure shows surgical instruments generally used for Thoracentesis procedures. We have illuminated surgical instruments from the front side whenever it is required.

for the algorithms throughout the entire experiments in order not to bias the results.

(iii) We used a 50 mL syringe to check our framework and to check the feasibility to use a template-matching algorithm for object recognition. We created 15 scenes containing multiple instruments, e.g. a syringe and a surgical-swab, and a table-top from different view-points. Using the "KinectFusion" algorithm, we obtained object templates.

We checked the instrument partial model's mapping with these scenes through template matching.

Table 2. Parameters of the plane segmentation algorithm.

Parameters	Details
Voxel grid size	It is used to adjust the point cloud density by specifying the size of 3D voxels in meters
Passthrough axis dimensions	Passthrough filter, filters the planes through X-, Y- and Z-axis with respect to camera frame
Maximum iterations	It is used to ensure that at the probability (usually set to 0.99) that at least one of the sets of random samples does not include the outliers.
Euclidean cluster distance threshold	Set the tolerance in meters for difference in perpendicular distance (d component of plane equation) to the plane to be considered the neighboring points the part of plane.
Euclidean cluster tolerance	Set the spatial cluster tolerance as a measure in L2 Euclidean space
Minimum cluster size	Set the minimum number of points that a cluster needs to contain in order to be considered valid
Maximum cluster size	Set the maximum number of points that a cluster needs to contain in order to be considered valid

The error metric — Huber penalty measure — hereafter also represented as fitness score, L_h was calculated as shown in Eq. (2).

$$L_h(e_i) = \begin{cases} \frac{1}{2} e_i^2 & \|e_i\| \le t_e, \\ \frac{1}{2} t_e(2\|e_i\| - t_e) & \|e_i\| > t_e. \end{cases} \tag{2}$$

In Eq. (2), e_i represents the error of alignment between the source and target point cloud. t_e represents the distance threshold that is used to limit the points whose values are greater than the specified value, which is sum of squared distances from two sets of correspondence distances between source and target points. We designed a paradigmatic test scenario to test the framework and to identify the pose of a large syringe (50 mL) from the scene, with model containing different point density, during the "Closure" phase, which is required during the withdrawal of the syringe to drain the fluid from the chest. In this case, the surgeon asks about the correct instrument, while reaching the "Closure" phase and the contextual information was retrieved by the system. Figure 7 shows the experimental setup which was used during segmentation and recognition of surgical instruments. We measured the execution time to identify system's requirement to be used in real time.

3. Results

Results are described in the next subsections.

(i) The ontology that we designed as an example of the Thoracentesis workflow is reported in Fig. 8. The procedure class for Thoracentesis is populated and it has relationships with steps, phases, surgical instruments, and body structures. Relations are formulated through the indicative propositions in ABOX. For example, as seen in Fig. 8, the class relationship between "Step" and "Instrument" is reported ("Withdraw" step has instrument aspiration syringe which is a 50 mL large syringe). We collected a list of instruments that are used during phases and steps of Thoracentesis procedure and populated the instrument class with instrument specific characteristics (e.g. instrument dimension, its 3D model instance, and so on). Ontology metrics are shown in Table 3.

As shown in Table 4, the average time to classify ontology is less than 1 s. The ontology consistency checking confirms that the ontology is syntactically valid and classification time confirms the ontology can be used as a surgical process model for context awareness in context-aware surgical systems.

For semantic analysis and ontological satisfiability, to check whether the ontology represents Thoracentesis, we run a total of 200 queries and we found a sensitivity of 100%, with specificity 99.1% and accuracy 99.5% of the knowledge representation. The specificity and accuracy are high because the procedure is based on a relatively small model and we used iterative and increment development approach for building the ontology. As shown in Table 3, it is not necessary to ground all individuals/instances of classes for the realization of Thoracentesis workflow.

(ii) We repeated the segmentation experiment for each instrument of Thoracentesis procedure using both segmentation algorithms e.g. RANSAC-based plane segmentation and color-based region growing segmentation, with white illumination and without illumination, for 12 times. Algorithms were able to segment surface patches for instruments as reported in Tables 5 and 6. The segmented surface patches were verified through visual inspection in ROS Rviz Module. As shown in Tables 5 and 6, we were able to get cluster point indices with less variability when we conducted experiments without illumination. However, RANSAC-based plane segmentation was unable to segment the object cluster for needles and a flexible catheter. The number of points represents precision or repeatability of

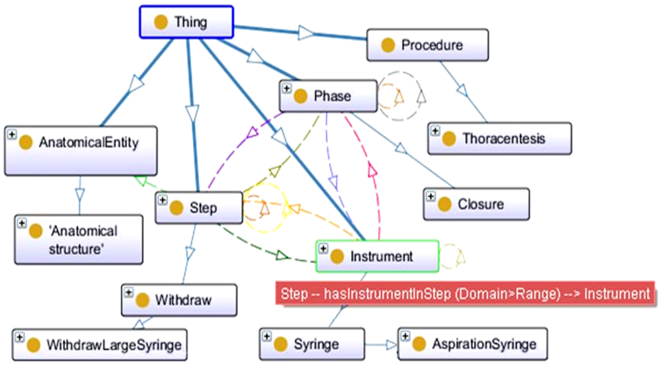


Fig. 8. The rectangles represent classes of the ontology, while blue edges represent hierarchies and other colored edges represent the relationship between the classes.

Table 3. Resulted ontology metrics.

Axiom	1805
Logical axiom count	569
Class count	105
Object property count	13
Data property count	18
Individual count	66
DL expressivity	ALCHIF(D)

Table 4. Ontology classification time during consistency checking.

DL reasoner	Classification time (ms)
FaCT++	83
Pellet	357

Table 5. Segmentation results without illumination (R represents RANSAC-based plane segmentation and C represents color-based region growing segmentation).

Instrument	Number of points	Execution time (s)
Syringe 50 mL	776.2 ± 166.0 (C)	4.72 ± 0.14 (C)
	915.9 ± 90.9 (R)	0.59 ± 0.62 (R)
Syringe 10 mL	379 ± 37.9 (C)	5.04 ± 0.19 (C)
	471.8 ± 96.3 (R)	0.78 ± 0.09 (R)
Surgical-Swab	7978.1 ± 2244.0 (C)	4.96 ± 0.15 (C)
	8655.5 ± 150.2 (R)	1.34 ± 0.10 (R)
Needle	444.1 ± 61.38 (C)	5.36 ± 0.28 (C)
Flexible catheter	2136.1 ± 86.51 (C)	5.57 ± 0.25 (C)

Table 6. Segmentation results with white illumination (R represents RANSAC-based plane segmentation and C represents color-based region growing algorithm).

Instrument	Number of points	Execution time (s)
Syringe 50 mL	1438.6 ± 0.18 (C)	4.48 ± 0.184 (C)
Syringe 10 mL	563.6 ± 98.2 (C)	4.52 ± 0.15 (C)
Surgical-Swab	727.18 ± 31.86 (R)	0.27 ± 0.01 (R)

Table 7. Plane segmentation results with respect to annotated ground truth with manual segmentation.

Instrument	Points (RANSAC-based plane segmentation)	Ground-truth model points (Manual segmentation)	Models' cloud-to-cloud distance (m)
Syringe 50 mL	4763.1 ± 602.3	6001.04 ± 2364.2	0.001 ± 0.001
Syringe 10 mL	1029.6 ± 247.5	1626.8 ± 327.9	0.002 ± 0.001
Surgical-Swab	7567.1 ± 486	7653.6 ± 408.6	0.0005 ± 0.0002

segmentation algorithms, while execution time represents amount of computational needs for processing required activity of the procedure.

As seen in Tables 5 and 6, RANSAC-based plane segmentation provides good results with less variability in a number of points and faster processing of the raw point clouds.

As shown in Table 7, RANSAC-based plane segmentation is able to efficiently segment the instruments. The repeatability is higher than the gold standard ground-truth model points, obtained using manual segmentation as explained in the experimental protocol, except for the surgical swab, which is also almost similar. We kept point cloud density higher for the reference clouds, e.g. ground truth models. Thus, the reference clouds are represented with high number of vertices that removes FP to cover the properties of an object we wish to compare for accurately measuring the distances between segmented models. As shown in ~~Table 6~~, model cloud-to-cloud distance, surgical-swab was segmented with the highest accuracy followed by 50 mL and 10 mL syringes.

We were able to segment syringes with the color-based region growing segmentation and surgical-swab with RANSAC-based plane segmentation when experiments were done under the white illumination. Neither of these methods did segment the other two instruments, needles, and flexible catheters, during experiments under white illumination. The reason was a light reflection from the instruments, which was creating missing depth information at a particular instrument space. The system heavily relies on the segmentation algorithm because recognition of instruments needs accurately segmented models. Moreover, we believe that 3D models would have allowed achieving results that are more accurate, however, in surgery when uncertainty is much higher, online segmentation of instrument is more appropriate to create context awareness. Higher repeatability means we had consistent segmentations, so the performance on recognition was consistent as well during recognition of instruments.

(iii) The instrument, e.g. a 50 mL syringe, could be recognized in the scene by template matching as shown in Fig. 9. In this experiment, we used RANSAC-based plane segmentation to segment the partial surface point-cloud model of the large syringe (50 mL) from the reconstructed scene obtained with the help of "KinectFusion".

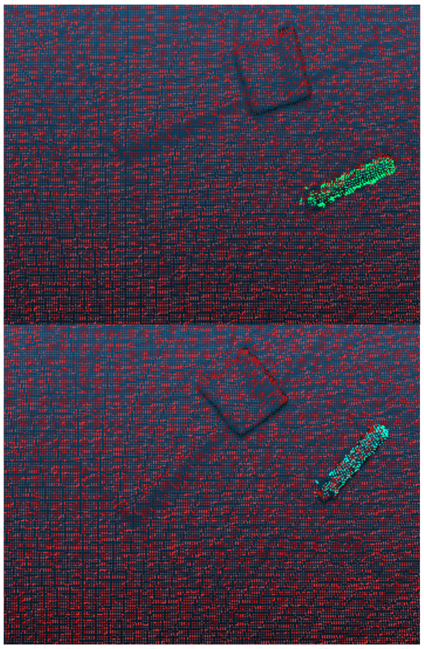


Fig. 9. Object recognition results. The light blue/green dataset is the partial surface point-cloud of the large syringe (50 mL) whose information (LargeSyringe.pcd) was obtained by the ontology. The red dataset, is representing two objects a syringe and a surgical swab on the table-top, alignment shows instrument, i.e. large syringe in the scene, in a specific context "WithDrawLargeSyringe". Two individual registered datasets are shown with red/blue and light blue/green colors, respectively.

Figure 9 shows the results of template matching of a large syringe in two different poses. The fitness score, i.e. the alignment result based on Huber penalty measure, based on threshold Euclidean distance value below 0.0020 mm indicates better template alignment with the scene images as reported in [34, 42]. We repeated this experiment with 15 different poses and the obtained results are listed in Table 8.

The results show the matching is more accurate with downsampled model at approximately 82% than with raw point cloud; however, further downsampling makes the results worse as reducing matching points

Table 8. Template matching results.

Template matching for an instrument	Execution time (s)	Average fitness score (mm)	Model's number of points
50 mL syringe – model 1 (obtained from downsampled point-cloud)	99.04 ± 6.4	0.020 ± 0.0053	109
50 mL syringe – model 2 (obtained from downsampled point-cloud)	96.65 ± 5.15	0.022 ± 0.0025	307
50 mL syringe – model 3 (obtained from downsampled point-cloud)	107.2 ± 20.6	0.016 ± 0.0047	771
50 mL syringe – model 4 (obtained from a raw point-cloud)	114.7 ± 10.2	0.019 ± 0.0043	4322

for alignment. While we achieved approximately 93.3% repeatability and approximately 93.3% accuracy in determining 50 mL syringe, the template matching had higher false positives, up to 80%, for the surgical swab and the 10 mL syringe verified by the visual inspection. The implemented template matching uses SAC-IA pipeline, which computes correspondences, based on normal vectors alignment. The template matching had less favorable results in the case of surgical swab due to geometrical issues.

The 10 mL syringe was too small so it was not possible to separate the syringe cluster from the plane. However, the results can be further improved by tuning some parameter, e.g. the search radius, which requires to search nearest neighbors and performing feature mapping between points. Moreover, the algorithm relies analyzing points' *XYZ*, i.e. position information only. It loses information about *RGB*, i.e. color information, which might aid in recognition. Our experiments with 50 mL syringe were promising as the syringe has bigger dimension and a lower surface curvature. Our experiments with 50 mL syringe showed that point sampling (down or up) does not affect execution time for the algorithm. However, we noticed that downsampling increases accuracy at certain level of approximation for recognition. The execution time is higher than expected, which limits its use during the real-time execution of the framework that can be improved by reducing point density of the scene. We also performed experiments during continuous real-time acquisition of the point-clouds but the results were impeded due to noisy acquisition of depth information.

4. Conclusions and Future Work

We presented a context-aware system, which we implemented using ontology-based framework. Context-aware systems are expected to enhance decision-making capability of the surgeons while operating in complex surgical environments, during robotic surgery for instance, where specialized knowledge is required to perform the procedure. The framework, which integrates ontology and data obtained using computer vision algorithms, has potential to create a context-aware system.

The highly configured system components have the potential to extend this framework to the laparoscopic surgeries, which can be robot-assisted. While laparoscopic surgery has special needs (e.g. complex instruments), our hardware profile, e.g. using Kinect, would be rudimentary. The approach could be enhanced by indirect sensor fusion, from vision, tracking, and imaging sensors, by grounding possible variables in the ontology. Indirect sensor fusion would be helpful to extend the support to existing robotic surgical applications e.g. selection of the best hardware to process specific data. When instruments are more unorganized in realistic environments, 3D lidar, stereoscopic cameras and optical or magnetic trackers would require accurately locating and tracking all the instruments for such procedures. Ontology along with imaging sensors, such as ultrasonography or computed tomography could enhance ontology-based intra-operative assistance, for example, to detect accurate body structure (e.g. needle insertion region on chest). As far as ontology is concerned, each surgical procedure has different steps, body structures, and different sets of instruments. It adds significant complexity and restrictions to the ontological scalability. The ontology could be scaled with OntoSPM [5] standard and state-of-the-art robotics ontology, such as Core Ontology for Robotics and Automation (CORA) [43]. The ontology can also be scaled by semantic mapping of meta-data, which is represented by ontological classes such as instruments characteristics. Meta-data can be included in the ontology as annotations for the ontological classes. Consequently, this common metadata could be used by different procedures to get information about similar instruments and thus removing architectural complexity to handle heterogeneous information. While our ontology is not extendable to such implementations, the basic design could improve with the inclusion of upper ontology such as Suggested Upper Merged Ontology (SUMO) [44]. The current ontology can be reusable within similar context only because the ontology lacks upper ontological axioms. The system performance was hindered by the limitation in detecting small and transparent instruments by Kinect. Another limitation was selecting optimal parameters for these algorithms; however, the approach can be solved by integrating parameters for each instrument in the ontology [45]. We found the fitness score below 0.020 mm during 14 experiments while template was not aligned properly with one experiment where fitness score was 0.029 mm due to the point-cloud obtained from improperly reconstructed mesh, which has holes surrounding syringe region as verified using visual inspection in ROS Rviz module. Template-based recognition algorithm might provide a different result during real-time processing however, it is also highly dependent on object features and its geometry. As edges can be a good representative for the object recognition, deep learning technologies could enhance the results with noisy datasets. The application-specific ontology contains linguistic variables such as "LargeNeedle", "SmallNeedle" and so on. Therefore, the system would not be able to understand task-specific needle size until the user provides the information explicitly through the queries during the run-time. Fuzzy extension to the ontology can handle uncertainty associated with such ontological classes and its assertions or indicative prepositions e.g. a linguistic variable and ontological entity "NeedleSize". The particular problem could be solved by developing fuzzy ontology where ontology provides the surgeon the best recommendation based on the rankings, constructed using conditional rules, for the instruments and hence it may improve the accuracy of the recognition system. Further to surgical training, the knowledge-based framework could be possibly extended to the intra-operative robotic assistance, e.g. robotic scrub nurse, where instrument's contextual information is forwarded to robotic scrub nurse and the robot will handle the instruments and assist surgeons during the procedure also detecting surgeon's intent [46].

References

1. R. Davis, H. Shrobe and P. Szolovits, What is a knowledge representation? *AI Mag.* **14**(1) (1993) 17–33.
2. N. F. Natalya and D. L. McGuinness, Ontology development 101: A guide to creating your first ontology, Stanford Knowledge Systems Laboratory Technical Reports KSL-01-05 and Stanford Medical Informatics Technical Reports SMI-2001-0880 (2001).
3. SNOMED-CT, USA: International Health Terminology Standards Development Organization, USA (2015), Available at www.ihtsdo.org/snomed-ct [accessed on 17 November 2015].
4. D. Katić, A.-L. Wekerle, F. Gärtner, H. Kennogott, B. P. Müller-Stich, R. Dillmann and S. Speidel, Knowledge-driven formalization of laparoscopic surgeries for rule-based intraoperative context-aware assistance, Lecture Notes in Computer Science, Vol. 8498 (2014), pp. 158–167.
5. D. Katić, C. Julliard, A.-L. Wekerle, H. Kenngott, B. P. Müller-Stich, R Dillmann, S. Speidel, P. Jannin and B. Gibaud, LapOntoSPM: An ontology for laparoscopic surgeries and its application to surgical phase recognition, *Int. J. Comput. Assist. Radiol. Surg.* **10**(9) (2015) 1427–1434.
6. S. Agrawal, A. Joshi, T. Finin, Y. Yesha and T. Ganous, A pervasive computing system for the operating room of the future, *Mob. Netw. Appl.* **12**(2–3) (2007) 215–228.
7. T. Neumuth and C. Meißner, Online recognition of surgical instruments by information fusion, *Int. J. Comput. Assist. Radiol. Surg.* **7**(2) (2012) 297–304.
8. J. E. Bardram, A. Doryab, R. M. Jensen, P. M. Lange, K. L. G. Nielsen and S. T. Petersen, Phase recognition during surgical procedures using embedded and body-worn sensors, *IEEE Int. Conf. Pervasive Computing and Communications (PerCom)*, Seattle, Washington, USA (2011), pp. 45–53.
9. Y. Kassahun, R. Perrone, E. De Momi, E. Berghöfer, L. Tassi, M. P. Canevini, R. Spreafico, G. Ferrigno and F. Kirchner, Automatic classification of epilepsy types using ontology-based and genetic-based machine learning, *Artif. Intell. Med.* **61**(2) (2014) 79–88.
10. L. Bouarfa, P. P. Jonker and J. Dankelman, Discovery of high-level tasks in the operating room, *J. Biomed. Inform.* **44**(3) (2010) 455–462.
11. C. Reiley, H. Lin, B. Varadarajan, B. Vagolgyi, S. Khudanpur, D. Yuh and G. Hager, Automatic recognition of surgical motions using statistical modeling of capturing variability, *Stud. Health Technol. Inform.* **132** (2008) 396–401.
12. T. Blum, H. Feussner and N. Navab, Modeling and segmentation of surgical workflow from laparoscopic videos, *Med. Image Comput. Comput. Assist. Interv.* **13**(3) (2010) 400–407.
13. E. De Momi and G. Ferrigno, Robotic and artificial intelligence for keyhole neurosurgery: The Robocast project, a multi-model autonomous path planner, *Proc. Inst. Mech. Eng. H: J. Eng. Med.* **244**(5) 715–727.
14. C. Broaddus and R. Light, *Murrey and Nadel's Textbook of Respiratory Medicine*, 5th edn. (Saunders Elsevier, Philadelphia, 2010).
15. S. Faruqi, C. Raychaudhuri, M. Thirumaran and P. Blaxill, Winging of the scapula: An unusual complication of needle thoracocentesis, *Eur. J. Internal Med.* (5) (2008) 381–382.
16. P. J. Baldwin, A. M. Paisley and S. P. Brown, Consultant surgeons' opinion of the skills required of basic surgical trainees, *Br. J. Surg.* **86** (1999) 1078–1082.
17. R. H. Flin, P. O'connor and M. Crichton, *Safety at the Sharp End: A Guide to Nontechnical Skills* (Ashgate, Aldershot, England, Burlington, VT, 2008).
18. H. Nakawala, E. De Momi, A. Morelli, C. Tomasina and G. Ferrigno, Ontology-based surgical assistance system for instruments recognition, in *Proc. CRAS: Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, Brussels, Belgium (2015), pp. 42–45.
19. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Ng, ROS: An open-source robot operating system, in *Proc. ICRA Workshop on Open Source Software.*
20. T. W. Thomsen, J. DeLaPena and G. S. Setnik, Thoracentesis videos in clinical medicine, *New Engl. J. Med.* **355**(15) (2006) e16.
21. Health On the Net Code of Conduct, Health On Net Foundation, Switzerland (2015), Available at https://www.healthonnet.orhg/HONcode [accessed on 17 November 2015].
22. T. Neumuth, G. Strauß, J. Meixensberger, H. U. Lemke and O. Burgert, Acquisition of process descriptions from surgical interventions, in *DEXA 2006*, LNCS, Vol. 4080, eds. S. Bressan, J. Kung and R. Wagner (Springer, Heidelberg, Germany), pp. 602–611.
23. Protége, Stanford Center for Biomedical Informatics Research (2016), Available at http://protege.stanford.edu [accessed on 12 January 2016].
24. A. Davidson, *Kinect Open Source Programming Secrets: Hacking the Kinect with OpenNI, NITE and Java* (McGraw-Hill/TAB Electronics, New York City, USA, 2012).

25. PCL - Point Cloud Library, (2015), Available at http://pointcloud.org [accessed on 17 November 2015].
26. A. M. Fischler and C. R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* **24**(6) (1981) 381–395.
27. R. Schnabel, R. Wahl and R. Klein, Efficient RANSAC for point-cloud shape detection, *Comput. Graphics Forum* **26**(2) (2007) 214–226.
28. Q. Zhan, Y. Liang and Y. Xiao, Colour-based segmentation of point clouds, in *Proc. XXXVIII ISPRS Congress*, Paris, France (2009), pp. 248–252.
29. R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davidson, P. Kohli, J. Shotton, S. Hodges and A. Fitzgibbon, KinectFusion: Real-time dense surface mapping and tracking, in *Proc. 2011 10th IEEE Int. Symp. Mixed and Augmented Reality*, Washington, USA (2011), pp. 127–136.
30. M. Gupta and G. Sukhatme, Using manipulation primitives for brick sorting in clutter, *IEEE Int. Conf. Robotics and Automation (ICRA)*, Saint Paul, Minnesota, USA (2012), pp. 3883–3889.
31. R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha and M. Beetz, Towards 3D point cloud based object maps for household environments, *Robot. Auton. Syst.* (2008) 927–941.
32. R. B. Rusu and S. Cousins, 3D is here: Point cloud library (PCL), *IEEE Int. Conf. Robotics and Automation (ICRA)*, China (2011), pp. 1–4.
33. M. Muja and D. G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, *IEEE Trans. Pattern Anal. Mach. Intell.* **36** (11) (2014) 2227–2240.
34. R. B. Rusu, N. Blodow and M. Beetz. Fast point feature hisotgrams (FPFH) for 3D registration, *IEEE Int. Conf. Robotics and Automation (ICRA)*, Kobe (2009), pp. 3212–3217.
35. B. DuCharme, *Learning SPARQL Querying and Updating with SPARQL 1.1* (O'Reilly Media, Massachusetts, USA, 2011).
36. H. Schildt, *Java: The Complete Reference*, 9th edn. (Oracle Press - McGrawHill Education, New York, NY, USA, 2014).
37. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne and K. Wilkinson, Jena: Implementing the semantic web recommendations, in *WWW Alt. '04:Proc. 13th Int. World Wide Web Conf. Alternate Track Papers and Posters*, edn. S. Feldman, M. Uretsky, M. Najork and C. Wills, New York, NY, USA (2004), pp. 74–83.
38. K. Walrath, M. Campione, A. Huml and S. Zakhour, *The JFC Swing Tutorial: A Guide to Constructing GUIs*, 2nd edn. (Addison Wesley Longman Publishing Co., Redwood City, California, 2004).
39. D. Tsarkov and I. Harrocks, FaCT++ description logic reasoner: System description automated reasoning, Lecture Notes in Computer Science, Vol. 4130 (2006), pp. 292–297.
40. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur and Y. Katz, Pellet: A practical OWL-DL reasoner, *Web Seman.* **5**(2) (2007) 51–53.
41. S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother and D. Kondermann, When can we use kinecfusion for ground truth acquisition? in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems Workshop on Color-Depth Camera Fusion in Robotics*, Vilamoura, Portugal (2012).
42. Aligning object templates to a point cloud – tutorial (2015), Available at http://pointclouds.org/documentation/tutorials/template_alignment.php [accessed on 25 January 2016].
43. E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, A. Chibani, S. Gérard, Y. Amirat and C. Schlenoff, Towards a core ontology for robotics and automation, *Robot. Autom. Syst.* **61**(11) (2013) 1193–1204.
44. A. Pease, I. Niles and J. Li, The suggested upper merged ontology: A large ontology for the semantic web and its applications, AAAI Technical Report WS-02-11, AAAI (2002).
45. H. Nakawala, G. Ferrigno and E. De Momi, Ontology-based instrument detection in a context-aware framework for surgical assistance, in *Proc. CARS 2016: Computer Assisted Radiology and Surgery 30th Int. Congress and Exhibition*, Heidelberg, Germany (2016).
46. F. Nessi, E. Beretta, C. Gatti, G. Ferrigno and E. De Momi, Gesteme-free context-aware adaptation of robot behaviour in human-robot cooperation, *Artifi. Intelli. Med.* (74) (2016) 32–43.

Hirenkumar Nakawala is a Ph.D. student in Bioengineering at Politecnico di Milano, Italy. He received his M.Sc. degree in Health Informatics from City, University of London, UK in 2013. He is working on the project of context-awareness in robotic surgery. His research interests include artificial intelligence in robotic surgery, clinical decision support systems, and healthcare interoperability.

Giancarlo Ferrigno received his Ph.D. degree in Bioengineering from Politecnico di Milano, Italy in 1990. He is currently a Full Professor and chair in Bioengineering division in the Department of Electronics, Information, and Bioengineering (DEIB), Politecnico di Milano, Italy. After completing Ph.D., he has worked as senior researcher in a private foundation from 1984 to 1990. He has been principle investigator of several projects funded by Italian Space Agency, Italian Institute of Technology, Industrial partners in the biomedical field and coordinated three EU funded projects in the 7th FP: Robocast(strep), Mundus(strep) and Active (IP).

Elena De Momi received her Ph.D. degree in Bioengineering from Politecnico di Milano, Italy in 2006. She is currently an Assistant Professor in the Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, Italy. She was a key participant of keyhole robotic neurosurgery (FP7 ICT 215190 ROBOCAST). She was a project manager and principle investigator for EUROSURGE (ICT 288233) project.