British Journal of Applied Science & Technology
14(4): 1-14, 2016, Article no.BJAST.23619
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN ***international***
www.sciencedomain.org

C-Semantic: A Novel Framework for Next-generation Robotic Vision via the Semantic Web Technologies

Alaaeddin Alweish[1*], Mohd Sharifuddin Ahmad[1], Alicia Y. C. Tang[1] and Azhana Ahmad[1]

[1]Universiti Tenaga Nasional (UNITEN), Putrajaya, Malaysia.

Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/23619
Editor(s):
(1) Samir Kumar Bandyopadhyay, Department of Computer Science and Engineering, University of Calcutta, India.
Reviewers:
(1) Aya M. Al-Zoghby, Mansoura University, Egypt.
(2) Dora Melo, Polytechnic Institute of Coimbra, Portugal.
(3) Georgios Stavroulakis, Technical University of Crete, Greece.
(4) Ming-Shi Wang, National Cheng Kung University, Taiwan.
(5) Wagner Tanaka Botelho, Federal University of ABC, Brazil.
Complete Peer review History: http://sciencedomain.org/review-history/13329

Original Research Article

Received 10th December 2015
Accepted 5th February 2016
Published 17th February 2016

ABSTRACT

Currently, research in robotic vision faces numerous challenges, predominantly because of noisy sensor input and the processor hungry practices of object detection. Conventional machine vision algorithms are unable to handle real-time scenarios efficiently because they mostly rely on local storage for objects and a limited training process. In real life, there are endless number of objects which requires a huge storage capacities and a high level of hardware to handle real-time images quickly. In this paper, we address the challenges of current robotic vision and propose a novel framework (C-Semantic) based on cutting-edge semantic web technologies. The framework divides the entire robotic vision process into three functional layers in which each layer performs a set of predefined tasks. The process begins with a vocal command that is further converted into a SPARQL query. We design a C-Semantic ontology that semantically stores the domain information along with objects' physical and geometrical features. The image-processing module of the framework receives an input image of an object and looks up for the object from the virtual environment by consulting the semantic features. An inference engine aids the image-processing

*Corresponding author: E-mail: alweish@hotmail.com;

module to rapidly detect and associate the object based upon the semantic relationships. Overall, the semantic powered kernel transforms the proposed framework into a robust, intelligent and interoperable system proficient to handle real-time scenarios. C-Semantic framework is evaluated against some scenarios from the literature. Based on the current experiments, the system displays favorable results. Based on our review, the integration of semantics with robotic vision algorithms is the first attempt of its kind that will pave the way for future research in this domain.

Keywords: Semantic web; robotics vision; NLP; SPARQL; object detection; ontology.

1. INTRODUCTION

Will robots help make life better? Will it really see, understand & decide one day!?

From earliest of times, artists conceptualized mechanical devices in the shape of humanoids. Later, these drawings took practical shapes when these sketches were transformed into real life by wooden structures that resembled humans. The prime motivation for inventing such structures was for entertainment purpose. Later on, these structures were called robots.

Robots have taken numerous shapes and sizes specifically suited to tasks that humans could not perform in light to heavy industrial processes. Advancement in robotic technology has been taking shape since the pre-industrial era and is currently capable of performing various functions that have revolutionized the automation process [1].

As time progresses, robots became more sophisticated and even wirelessly controlled. Currently, robots are utilized in various industrial applications without retaining the humanoid perspective in order to integrate robotic systems with industrial processes [2].

One of the most important components of a robot are the sensors. Sensors are input devices like a camera that provides vision for a robot. The use of sensors to hear, touch and move is also utilized to manifest human-like qualities and capabilities [3]. Vision is achieved through cameras and sonars and the resulting images are processed using complex algorithms.

The touch sensory feature for a robot is achieved through the robot's own processing of its movements. These movements are then utilized to create the desired movements and to attain the desired objectives. One such example is achieving a balanced state of physical movement which is a typical feature of humanoid robots [4].

Robots sensors gather data from the surrounding environment, then process it and take the appropriate actions depending on robots' goals and abilities. The main challenge in robotics vision is how to gather the information correctly and interpret them?

In order to resolve these challenges in the realm of information technology, computer researchers and programmers have developed semantic interoperability, by which computers 'understand' the meaning of data. Semantic interoperability works by adding metadata in-line with actual data and analyze this data by connecting each data element with a suitable ontology term. The performed process primarily depends on special data structures such as XML and RDF [5].

The structures of XML and RDF are linked with software to share data and capable of creating links with external software resources. The shared bank of vocabulary that indirectly links with the ontology basically provides the infrastructure that supports the interpretation, analytics and logic creation of the computer system, which manifests the Semantic Web [6].

Using techniques that incorporate XML codes defines the semantic philosophy, which further facilitates the analysis of web contents. The RDF system is responsible for arranging the contents into special triples. The object logic is created in the process as the RDF system works. Due to its varied usefulness, many industries are currently employing the RDF tool [7].

In this paper, we address the challenges of current robotic vision which includes: i) The weakness of machine understanding and ii) The limitations in robotic vision and objects recognition.

The current machine and robotic vision suffers from:

- **Disunion in objects databases:** there is no shared datasets and no standard classification environments for object recognition.

- **Differences in the recognition approaches:** the main challenge of objects recognition is to reach the maximum level of accuracy with the minimum possible time. Most of the current approaches rely on traditional images processing, which is relatively slow and needs a lot of training.
- **Image training:** This requires a massive amount of data storage to store each object in many positions.
- **Objects are getting complex:** with the advancement of web approaches, objects representation is getting more complex with more relations which are getting more complex.

The objective of this study is to make the robotic vision faster and more intelligent by enhancing their ability to realize objects, their relations and properties.

The contribution of this work is a shared framework and an API for machine vision that gives the ability to access graphs of objects that are shared on the internet without the need to use a database. All of these components are open source and available for researchers and developers.

The rest of this paper is organized as follows: In Section 2, the related work is presented. Section 3 elaborates the proposed framework in detail. Section 4 explains the experiments and discusses the problems that we encounter during the experiments. The paper concludes in Section 5 with a proposal for future work.

2. RELATED WORK

Prestes et al. [8] presented the Service-oriented Ubiquitous Robotic Framework (SURF), an idea to improve robotic intelligence. The SURF technology basically incorporates a service-oriented approach into special stimulations and environments to achieve the final objective of networking robots. SURF relies on semantic technologies for interlinking robots and computing devices. This is a revolutionary idea that has the capacity to inspire the next generation of advancement in robotics and interlinking of computing devices.

Research work on evaluation of ontology has taken many shapes and is an active subject of consideration when we talk about robotics and its interlinking of networking hardware devices. The research in this area is suggestive of the fact that METHONTOLOGY can be a method to further define ontology. Therefore, advanced research has been conducted in the fields of robotics and interlinking of service devices through the method mentioned above [8].

Further researches on robotics suggest that Ontology systems can also rely on Microsoft Robotics Studio (MSRS) simulations as an implementation framework. When applying robotics to service-oriented sectors and technologies, MSRS system can be used successfully in the specialty of robotics to achieve many goals. These goals include developing faster processes with the benefit of in-depth integration of diverse ontology systems [9].

The most beneficial aspect of MSRS is that it incorporates a Service-Oriented Architecture (SOA) simulation software, thus transforming robotics simulations while keeping within the limits of ontology-based approaches. When exploring the topic of cognitive robotics, it is important to develop a model that is based on the SOA and related to ontology. SOA technology provides a verification approach for the robotic system. Furthermore, this system ensures that the ontology-based system that is in-line with SOA framework is a real possibility when it comes to cognitive robotics [9].

In [10], a fuzzy logic based robotic vision system is presented in order to identify the real-time less precise objects with accuracy. The main contribution of this research work is to provide a state based system to maintain context that works with fuzzy logic controller to imitate the low level imprecise reasoning of humans to abridge the vision tasks for a common serving robot. The usage of fuzzy logic helps a robotic vision system to process vague images. Moreover, a fuzzy based robotic vision system enables the usage of less complex image processing operations and algorithms.

These characteristics of fuzzy based robotic vision system make it favorite to use in real-time object detection. In order to detect the indoor obstacles, a linear structured light vision system is presented in [11]. This research work simplifies the calibration process by introducing a coordinate system based on structured light vision system.

Researchers in [11] have employed 650 nm light filter in front of the camera lens to project a structured light that facilitate a robot vision system to gather the environment information. Vision system keeps records of all objects along with their local coordinates. This approach assists to detect the change of linear structured light per frame. This information uses an image processing algorithm to understand the obstacle characteristics such as size and dimensions. Researchers claim that their experiments describe the linear structured light vision system is the most appropriate system for mobile robot obstacle detection in indoor environment.

To improve the robotic vision system in 3-D image application, [12] introduced a 3-D Time-Of-Flight system (TOF) that works in integration with robotic vision system. This system helps to correct segmentation of objects with equal gray-level values unlike stereo vision system that face difficulties to find the relevant image coordinates.

In [12], S.Y. Chen briefly surveys the recent developments for robot vision by using Kalman filters. Kalman filters have more than fifty different variations and most actively used to resolve the issues relating to solve uncertainties in robot localization, navigation, following, tracking, motion control, estimation and prediction, visual serving and manipulation, and structure reconstruction from a sequence of images. The major contribution of S.Y. Chen [12] works to provide the idea of vision localization that has proved itself while experimentation a reliable method to detect the complex objects.

A single point marker technique is introduced in [13] to improve the assembly process. An accurate robot camera calibration plays a key role in the whole process. Therefore, researchers in [13] have focused on the intrinsic camera and hand-eye calibration on a robot vision system using a single point marker. The single point markers make the object calibration process simple and a user do not require bulky special purpose calibration objects. More of this technique, it accelerates the on line accuracy checking and re-calibration when needed, without altering the robots production environment.

Seung-Ho Baeg in [14] proposed an object recognition system for a robot-assisted future home environment based on RFID tags and Visual Descriptors. In that system, every product is registered through visiTag, software for object recognition based on RFID visual descriptors extraction. The system is supported with an annotation tool that generates a variety of visual descriptors of MPEG-7 specification in the extensible markup (XML) and stores the descriptors into their own Object Information Server (OIS). Those description data include colour, texture and shape descriptors and is used for the matching processes. The robot gets the RFID code of the object and sends it to an Object Naming Server (ONS) which maintains the addressed information about which OIS maintains the object data currently requested.

Priyamvada Singh, in [15], created a domain ontology with contextual and image features to retrieve and generate a list of URI’s of the Electronic Health Records (EHR) of patients. They combined this ontology with the concept of CBMIR (Context Based Medical Image Retrieval) which is only specific to medical images, and uses only the low level features with the SPARQL as a query language to retrieve the results from the ontology. The image features class deals with the image properties such as the color distribution, color histogram and the region of interest, while the features in the contextual ontology class are based on the context of the data such as the dependent, partially dependent and the independent features. The used semantic approach showed better and more precise results as compared to the normal keyword based search.

Visual information contained in a document may not be reachable due to lack of adequate or proper textual and or low level image descriptor, to address this problem Alberto Chavez-Aragon [16] introduced Image Retrieval by Ontological Description of Shapes (IRONS), he suggested that low level image descriptors does not have a semantic value and cannot satisfy users query intention accurately, he used an ontology that describe a domain of shapes with image database in which images are in specific categories, his method is to extract the simplest shapes from an image then matches these shapes with the previous ontology, Test result indicated the accuracy decreased as the number of image increased ranging from 40% up to 60%.

The subject of robotics and its functional dependence can also be tested with a broader array of ontology related to designing of systems for the efficient working of robotics. Researchers in [17] employ ontology-based approaches to align robotics in humans’ daily lives with the help of open robot ontology concepts that primarily achieve the objective of successful functioning of robots.

As the field of ontology of robotics is further explored, robotics and mobility can be validated and explored with systems dependent on structured architecture. The structured architecture is derived from knowledge-based systems and tools. This architecture when implemented on a wider level is known as the global architecture of ontology of robotics [18].

Advanced researches on robotics ontology also incorporate the process to avoid collisions by assessing collision risks and collision avoidance protocol. These researches have contributed greatly to the generation of current state of robotics, which can perform diverse useful functions [19]. It is a known fact that specialized ontologies are required for very specialized robotics. For example, in underwater vehicles research, researchers have reported that Support Vector Machines (SVM) have been developed to recognize images which eventually allows unmanned underwater vehicles for use in exploratory services [20].

Furthermore, when considering the strides in the development of humanoid robots, the ontologies related to robotics have been developed in such a way that robots can recognize gestures and voices. The gesture detection in robotics is implemented through a Gaussian matrix model. This is a great sophistication in the robotic life cycle [21].

3. THE PROPOSED C-SEMANTIC FRAMEWORK

We coin the term C-Semantic to denote the operation of the framework in which the letter ‘C’ represents ‘see’ for vision and ‘Semantic’ represents the technology for seeing. In proposing the framework for intelligent robotic vision, we divide the C-Semantic architecture into three functional layers.

In Layer 1, a sensor receives a vocal command, (e.g., find a remote TV controller), from an operator, which is then processed by a Natural Language Processing (NLP) module. The result of the NLP module yields a formalized query which is transformed into a SPARQL query. The framework matches the SPARQL query against the C-Semantic ontology that concurrently links semantic rules and the image processing module of the system.

In Layer 2, an inference engine is employed that infers the commanded object by generating a virtual semantic map of the object’s physical and geometrical features. The final result is stored as a triple along with the RDFized properties and relations.

Layer 3 provides access to the system in three ways based on user type and competency. A developer can access the C-Semantic API and can utilize the features to develop his/her own application. A researcher who works on semantic integration projects can utilize the SPARQL endpoint to get a federated view while a naive user can access the system through a semantically driven interactive interface.

Fig. 1 shows the three layers of the C-Semantic framework. The sections below explain the internal working of each layer.

3.1 Activities in Layer 1

3.1.1 Sensor interaction utility (SIU)

The sensor interaction utility provides a vocal command interface for an operator. Upon receiving the voice input, a microphone passes the signal to a machine perception algorithm that performs the preprocessing on voice signals by removing the noise and by adjusting the sound pitch. The intelligent algorithm automatically adjusts the sound according to defined criteria and asks for repetition in case the sensor failed to recognize the command.

SIU handles both action commands and query commands. Query commands are commands with which an operator requests to locate something, e.g., locate the refrigerator in the room, while the action commands refer to the commands that control the actions of robots such as: Start search, stop, set camera directions, set directions, etcetera.

3.1.2 NLP processor

The sensor interaction utility forwards the refined input to NLP module as Natural Language Command (NLC). In Layer 1, the system decomposes NLC into small data chunks and converts it into textual format. We employ the Google Text to Speech API along with the Natural Language Tool Kit (NLTK).

NLTK is a freely available natural language processing framework that uses the WordNet and other artificial neural networks in its background to optimally process a natural language input. Fig. 2 shows the steps of NLP stage in C-Semantic.

The lexical analysis stage performs some advanced functions on the textual input. For instance, a stemmer algorithm resolves each query word into its root; words like moving, moved and mover are resolved to its root word '*move*'. The tagger component takes each word and segregates it into categories such as noun, verb, linking clause and etc. The Stop-word reduction discards special component in lexical analysis; it removes all the articles and unwanted prepositions and auxiliary verbs including is, am, are, the, of, at and etc. Finally, the parser function parses the resulting texts and generates a random query. The output of this function forwards the query to the query formulation module that verifies the lexical order for further processing.

3.1.3 SPARQL query generator (SQG)

SPARQL query generator module takes the formulated text query and transforms it into a SPARQL query [22]. SQG algorithm processes the text and segregates the classes and objects and properties from the query by consulting the WordNet and semantic terminologies stored in C-Semantic ontology. We store several commands and alternate commands in the ontology by using '*sameAS'* semantic mapping. For instance, the basic command to execute a query is 'run', however the natural language interface allows a user to choose alternate words to ask a question.

If a user ask 'Please find the white ASTRO remote control', as shown in Fig. 3. The word *'the'* and *'Please'* are removed; *'the'* is an article and *'Please'* is a sentence starter. We have manually created an ignore-words list that allows the algorithm to remove those words from the natural language input at the time of processing. However, words like 'Please' that can also be used as a command like, 'Could You Please' is retained by allowing the algorithm to annotate Such words with the '+' symbol, which indicate that these words are part of query and should not be removed. Consequently, NLP algorithm takes care of those words and does not remove such words but instead the algorithm looks up for an alternate word which could formulate a more optimal query to process.

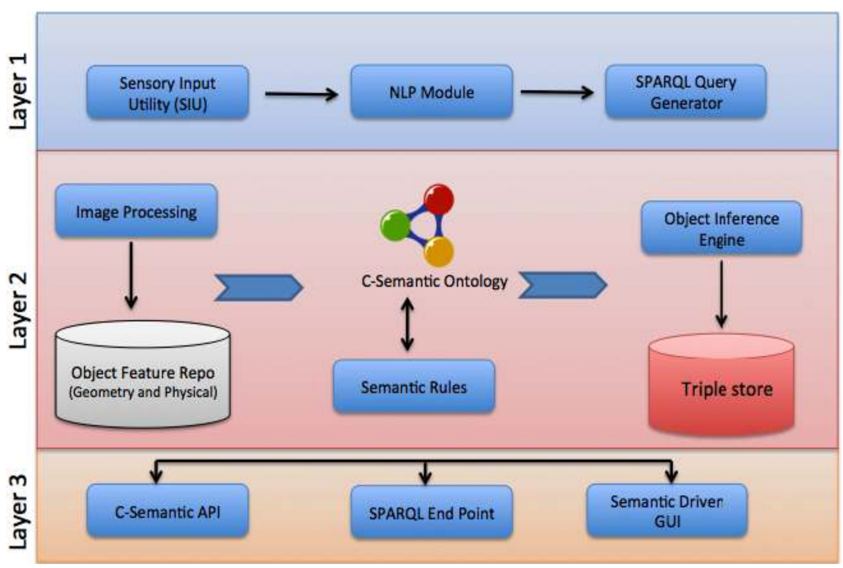


Fig. 1. The graphical framework of C-Semantic

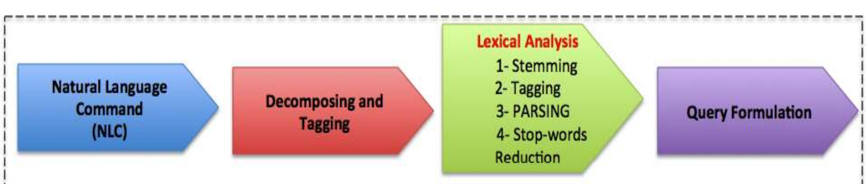


Fig. 2. Natural language processing stages in C-semantic query parser

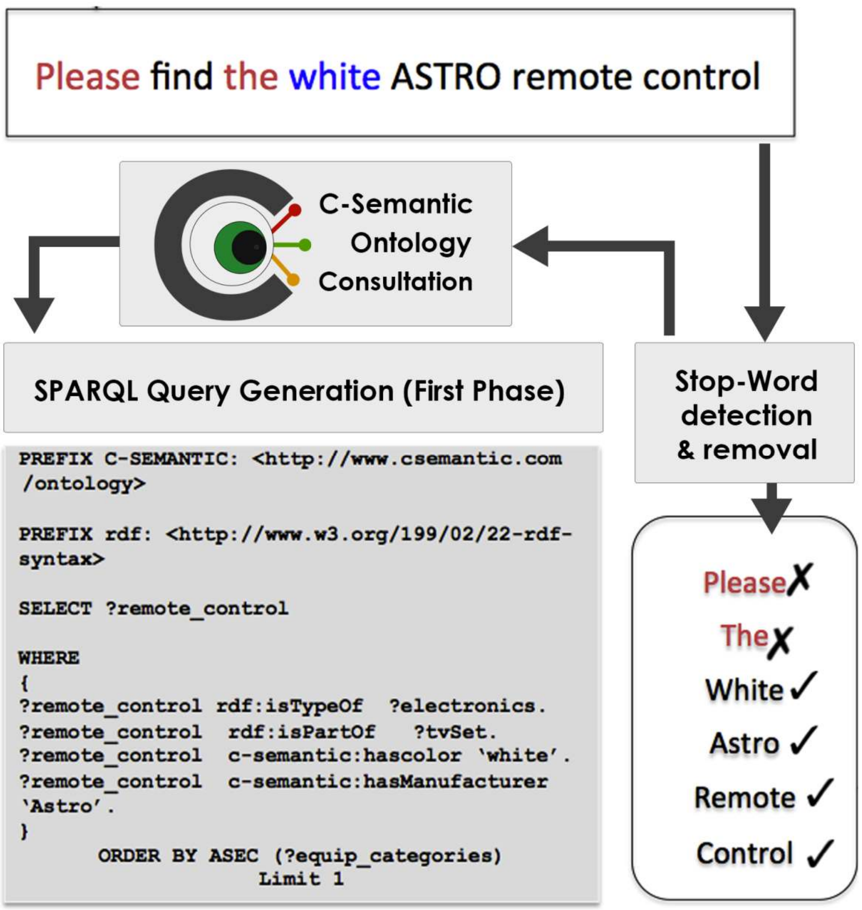


Fig. 3. SPARQL query generator

3.2 Activities at Layer 2

3.2.1 Image processing and object detection

The image processing module performs all the necessary tasks that are required to get fast and accurate detection of an object. Therefore, the image-processing module is linked directly with C-Semantic ontology, which is explained in section 3.2.3; image features repository and a rule and inference engine. After getting the SPARQL query from the first layer, the framework passes the output to the query manipulation function that executes the query against the C-Semantic Ontology. The C-Semantic ontology holds domain information that is semantically stored and returns possible features of an object.

The image processing utility works in stages. It performs the image pre-processing in the first stage; when the image processing utility receives an image from the camera, it checks for image resolution, increases the resolution and adjust the layout, if required. We have preset a threshold limit for image resolution and scaling in the code.

In the second stage, the image-processing engine connects itself with the object feature repository and constructs a virtual environment to precisely recognize an image. For instance, if the algorithm finds a door lock or handle in the sensor input, it tries to construct a set that adds the door and surrounding walls and other objects in it. The C-Semantic ontology stores the hierarchical information about a searching environment. Therefore, it could infer related information for example; it could infer walls, a roof, door and a floor if a query comes with word *'room'* in it.

In the third stage, the image processing module detects an object and registers its updated features in the object feature repository and in the ontology. We utilize the OpenCV to process the images. OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms [23].

3.2.2 Object feature repository

The object feature repository is a special kind of repository that holds images and their related metadata such as their physical and geometrical features. Moreover, to increase object detection precision and to observe the efficiency of the algorithm, we add trained datasets of various domains. The physical features about an object includes, length, height, width, color, object nature, e.g. liquid, solid and etcetera, while the geometrical features cover the position of the object with reference to another.

The object feature repository also holds a corpus that stores the related object. Whenever a user queries to find a particular object and the robot discovers it correctly, a log file is generated along with the associated objects. Related object corpus read the log file and populates itself with related images. The self-learning approach saves object searching time and exceptionally increases the precision.

3.2.3 C-Semantic ontology and rules

Ontology can be explained as a branch of metaphysics that is primarily focused on actual essence and existence of an entity. Ontology can be further explained by dynamic specifications of shared idealism or conceptualization of a domain [24, 25].

C-Semantic framework utilizes the ontology concept in which hierarchical order is used to form subclass and superclass hierarchy with a primary focus on concepts of domains. The primary motive to develop an ontology system is to facilitate a platform that allows mutual understanding of the knowledge content in the domain. This purpose can be achieved with the help of software. The software basically utilizes varied classes of domains instead of remodeling the data.

The ontology concept is dependent on modeling that uses special programming language called OWL [25]. OWL stands for Ontology Writing Language. There are various versions of OWL that include the OWL 2. OWL 2 has improved features such as better meta-modeling features for increased elaborative expressivity. These extended features basically facilitate the powerful modeling capacity of OWL 2 systems. Furthermore, there are several ways that have been employed to generate OWL 2 which make OWL 2 comprehensive data mining software with enhanced features [24].

Several theories also surround the ontology development process that is comprehensively defined in the seven-step process:

- (i) The recognition of the scope and domain of Ontology.
- (ii) Reutilization of older versions and existing Ontologies.
- (iii) Decoding and enumerating process description in Ontology.
- (iv) Defining the steps and hierarchical structure for Ontology.
- (v) Defining the various classes in detail.
- (vi) Defining the slots and facets.
- (vii) Create various instances

We develop the C-Semantic ontology to semantify the process involved in object detection and recognition. The semantic knowledge about objects minimizes the object detection time and system resources. We integrate the semantic web rules with our ontology that gets the object feature information from object feature repository and facilitates the system to infer based on semantic knowledge instead of using the conventional training and testing methods for object detection and recognitions that have been used for decades.

We build our ontology in Protégé, a free, open-source ontology editor and framework for building intelligent systems. Protégé is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organizational modeling [26]. Protégé add rules through the SWRL tab. The Semantic Web Rule Language (SWRL) is a language that provides the necessary logic to ontology. The SWRL recommendation was submitted in 2004 by the National Research Council of Canada (NRC).

Based on the logical feature, a user can retrieve inference information out of ontology. SWRL is considered as powerful as OWL DL but at the price of decidability and practical implementations. For example, if we arranged a doorbell and handle as parts of a door, when a robot is commanded to find a door from similar objects that look like a door, a door lock or bell helps the inference engine to deduce that the object is a door based on its association with doors and surrounding objects [27].

3.2.4 Object inference engine and triple store

We utilize the 'Pellet inference engine' [28] to generate inference in order to enhance the robotic vision. The inference engine connects itself with the ontology and the SWRL rules upon receiving a query and suggests the possible results to the user. An important feature of inference generation algorithm is self-optimization; the algorithm learns from its history and utilizes the saved knowledge to generate new results.

The framework RDFizes the object's physical and geometrical data along with its actual object references and stores it in the triple store. We utilize the Sesame Triple store in our experiments to store the semantic knowledge. Sesame is a powerful open source framework for processing and handling RDF data. This includes creating, parsing, storing, inferencing and querying over such data. It offers an easy-to-use API that can be connected to all leading RDF storage solutions [29]. We develop a number of web services that perform the transportation and display functions. An inference engine or an image processing unit can ask for stored RDFized data anytime while processing a request [30]. A web service that connects with the triple store pulls out the requested information from the repository.

3.3 Activities in Layer 3

The C-Semantic API is a collection of reusable classes and methods that can be accessed and utilized to develop an application when an improved computer vision usage is required. Moreover, we provide the SPARQL endpoint for users who are interested to explore the process of semantic vision. Through endpoint, a user writes a query in the SPARQL language and the application generates the results in RDF format, which could be transformed further into XML, JSON to directly utilize it for analysis purpose.

For naive users and multidisciplinary researchers, we develop a graphical interface, with which a user sends commands to accomplish a task through a robot or simply performs the semantic vision related experiments.

4. EXPERIMENTS AND RESULTS

C-Semantic is evaluated for (1) Object recognition performance and (2) Object realization features.

We tested our system on a scenario named: "Find a Door". In this testing scenario, we evaluate a real-life recognition task, compare the methodology and testing results of a related work with ours, and describe how C-Semantic features enhance machine intelligence by making objects not only recognized but understandable and realizable. We are using our custom-made robot in a real-life scenario where the robot is in the middle of a corridor, a hall or a connecting room. And the task is to find a door/doors.

The generic characteristics of a door make it very difficult or even impossible to be recognized in a real-life scenario. The majority of a door shape is an empty space on the wall which is also an empty space in with no unique attributes to be recognized. Bailey et al. [22] presents a robotic vision system which is based on fuzzy logic. This system is tested on one scenario which is "door locating in a corridor". Their robot detects doors by reading the nameplate of each door and then locates the door edges.

Their proposed system in [22] is implemented using MATLAB with a large number of input images for the corridor from different angles. Their robot is able to detect doors in a corridor only when the robot is aligned in the middle of the corridor. According to Bailey et al. [22], they still need to develop the alignment mechanism. And no any numerical or graphical representation of their results is provided. Fig. 4 shows the State transition diagram for searching for a door.

The methodology of finding a door using C-Semantic is more flexible, where it is not necessary to have a nameplate or to align the robot in the middle of a corridor.

In our system, the process start with a vocal command which is in our scenario "Find all doors", the speech-to-text API converts this vocal command into a text command. The output text is passed as an input to the NLP unit for processing which passes the result to the C-Semantic engine which prepares the command "Find All doors" along with the NLP analysis to the Order Extraction unit, Semantic Reasoning unit, Domain Selector and Ontology Parser respectively in order to create the SPARQL query and execute it on the C-Semantic graph which in turn cooperates with linked data unit and locates the triple of our object "Door".

Fig. 5 shows the SPARQL query result, it became ready in 4 seconds on 1Mbs internet connection speed.

By finding the triple of the "Door", C-Semantic will have a powerful access to unlimited number of "Door" characteristics, attributes and relations, and doing this before starting the image processing functions will totally change the process of object recognition. The knowledge gathered from the graph in Fig. 5 manages the image capturing and processing of our robot.

From the C-Semantic graph, the robot extracts the Shape Style of the "Door" which is a "Rectangle" as shown in Fig. 6a, for shape style operations we start with a simple image processing for "Edge detection" as shown in Fig. 6b.

The image processing module builds array of rectangle objects, and starts a multi-threading operation looking for the related objects recursively. Each abstracted object is linked to many Image triples, where each triple contains a URI for a physical image stored in the internet. Before starting the regular image recognition, the strategy of C-Semantic is to obtain as much knowledge as possible for the related object, characteristics and attributes, the knowledge obtained supports the recognition process by minimizing the image search criteria which leads to faster processing time comparing with conventional image training recognition approaches. As shown in the extendable graph in Fig. 5, the objects related to a door are "Hinge", "Closer" and "Handle".

Recognizing a "Handle" for example either by recognizing its related object the "Lock" recursively, or using the image processing module to loop over the physical linked Images of the "Handle" as in Fig. 7, object will finally lead to recognize the "Door".

We applied this scenario 8 times in a corridor with 4 doors. In each experiment, the doors state was different. Table 1 shows the environment and results of each experiment.

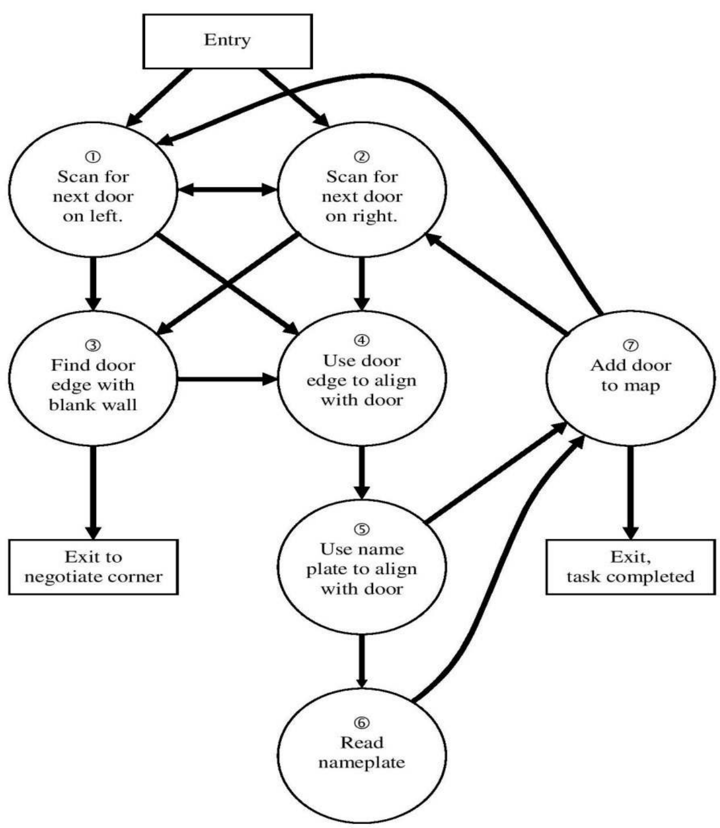


Fig. 4. State transition diagram for searching for a door [22]

We executed the above task in different scenarios and stored the results of each experiment. Equations 1 and 2 are used to calculate the detection precision and recall [31]. Table 2 highlight the precision and recall ratio derived from the experiments.

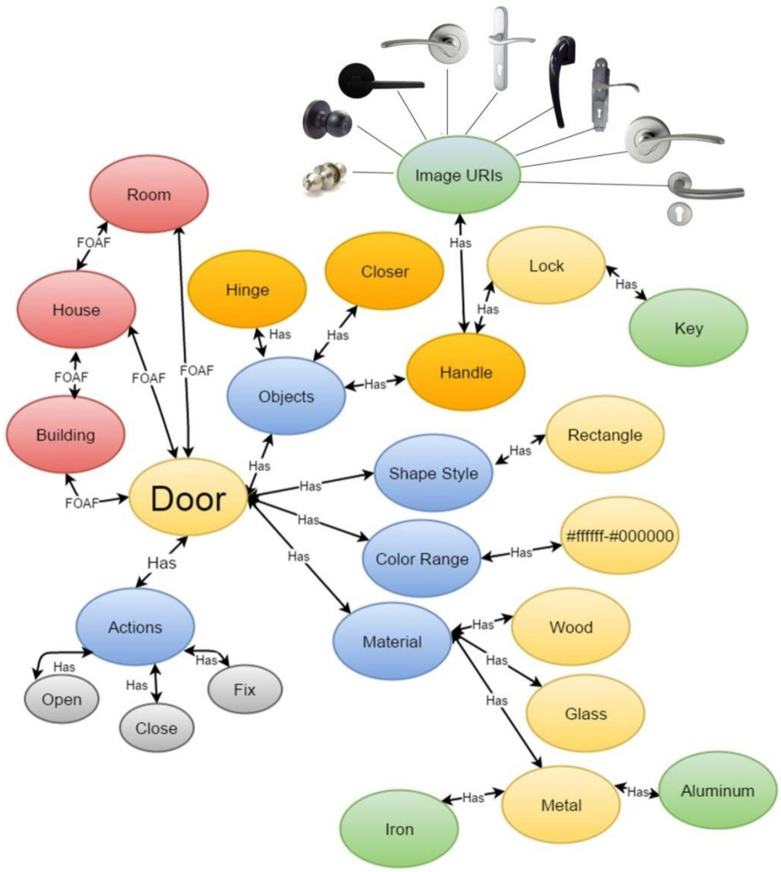


Fig. 5. SPARQL query result for door scenario

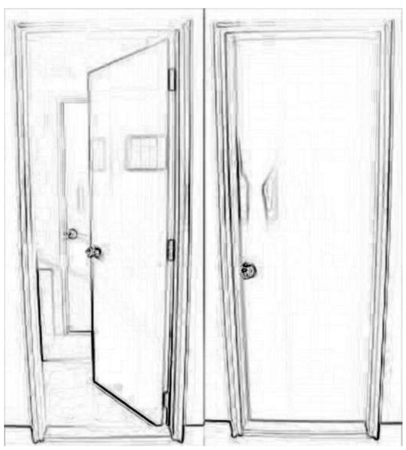
Table 1. C-Semantic testing experiments

No	Environment	No of correctly detected doors	No of false detected doors	Query time (S)
1	4 closed rectangle doors with standard handle	4	0	3.4
2	4 closed doors: 2 have a handle in a shape that is slightly different from the images linked to the “Handle” object. The other 2 doors has a standard handle	3	1	3.1
3	4 closed doors: 3 doors have a standard handle with no key lock. 1 door has standard handle shape with key look	4	0	3.5
4	4 closed doors with handles that have the same color as the door but with different lighting and shadow.	2	2	2.9
5	4 open doors and their handles are in robot vision range.	4	0	2.8
6	4 opened doors.2 doors are in the vision range and 2 are only opened to the opposite side of robot vision range.	3	1	3.6
7	4 closed doors. 2 open doors with handles in robot vision range. 2 handles have the same color as the door.	3	1	3.8
8	4 closed doors without handles, only key lock.	3	1	3.4

Table 2. Average result recorded with C-semantic robotic vision system

	TNOD	TCOD	TFOD	Precision (%)	Recall (%)	QET (Sec)
Find a door 8 scenarios	32	26	6	81.25	81.25	3.3

a) Original doors image



b) Doors image after processing

Fig. 6. Image of doors in a corridor before and after processing

Precision = the number of objects correctly detected as a percentage of the total number of object detected.

$$Precision = (\mathrm{TCOD} \,/\, TNOD) \times 100 \qquad (1)$$

Recall= the number of objects correctly detected as a percentage of the total number of object targeted.

$$Recall = (\mathrm{TCOD} \,/\, TNOT) \times 100 \qquad (2)$$

Where:

TNOT = Total number of Objects Targeted
TNOD= Total Number of Object Detected
TFOD= Total Number of False Object Detected
QET = Query Execution Time

The results of the above scenarios show the high efficiency of our proposed system. The only cases that C-Semantic robot could not find is the cases of not finding any object that is related to a door.

Fig. 7. Door handle

5. CONCLUSION AND FUTURE WORK

In order to overcome the robotic vision challenges, we present in this paper a novel framework that incorporates the state-of-the-art semantic web technologies to make robotic vision intelligent and efficient. The robust architecture of the C-Semantic system empowers a robot to accurately detect an object based on its semantic features. The semantic web backed system turns the overall robotic vision process into interoperable smart semantic activities.

C-Semantic system provides a multi-modal access to the system. With C-semantic API, a developer can access the system programmatically. To perform semantic dataset integration and federated queries activities the system can be connected through the SPARQL endpoint. For researchers and for non-domain users, C-Semantic system facilitates a graphical user interface.

We design scenario-based evaluation processes to monitor the performance of the overall system. With initial experiments, the system generated favorable results. We face a number challenges

due to resource-hungry nature of robotic sensors, however, the efficient distribution of resources due to our layered approach solved those challenges. We can say that with the current setup, our system is capable for adoption in many commercial products.
The next version of C-Semantic should mainly focus on upgrading the API components. C-Semantic API needs more development for the modules of data entry validation, error handling and data reliability, although those modules are seems not to be playing a major role in the framework's functionality. However, some validation functions like non-duplication validation might influence the main framework functions if not implemented accurately, the object "Door" for example should not be added in the graph more than one time for the exact same purpose.

C-Semantic API also needs more development for better support of knowledge and experience sharing. The framework however is well-designed, analysed and totally flexible to host the scenarios of knowledge and experience sharing.

ACKNOWLEDGEMENT

Sincere thanks goes to all members of the Centre of Agent Technology (CAT) at Tenaga National University (UNITEN) for sharing intelligent ideas. Special thanks also goes to the members of Stackoverflow.com for sharing technical solutions.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Livingston CD. Introduction to robots. Doctoral dissertation. Virginia Military Institute.
2. Nolfi S, Floreano D. Evolutionary robotics. The biology, intelligence, and technology of self-organizing machines. MIT Press; 2001.
3. Kemmotsu K, et al. Recognizing human behaviors with vision sensors in a network robot system. in Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. IEEE; 2006.
4. Hirose M, Ogawa K. Honda humanoid robots development. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. 2007;365(1850):11-19.
5. Wang X, Gorlitsky R, Almeida JS. From XML to RDF: how semantic web technologies will change the design of'omic'standards. Nature Biotechnology. 2005;23(9):1099-1103.
6. Klyne G, Carroll JJ. Resource description framework (RDF): Concepts and abstract syntax; 2006.
7. Riad M, Elminir K, Abd-Elghany S. A literature review of image retrieval based on semantic concept. International Journal of Computer Applications. 2012;40(11):12-19.
8. Prestes E, et al. Towards a core ontology for robotics and automation. Robotics and Autonomous Systems. 2013;61(11):1193-1204.
9. Tsai WT, et al. An ontology-based collaborative service-oriented simulation framework with Microsoft Robotics Studio®. Simulation Modelling Practice and Theory. 2008;16(9):1392-1414.
10. Bailey D, Wong YW, Tang L. Hierarchical fuzzy state controller for robot vision, in Robot Vision. Springer. 2008;391-402.
11. Wei B, et al. Indoor mobile robot obstacle detection based on linear structured light vision system. in Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on. IEEE; 2009.
12. Chen S. Kalman filter for robot vision: A survey. Industrial Electronics, IEEE Transactions. 2012;59(11):4409-4420.
13. Lundberg I, Bjorkman M, Ogren P. Intrinsic camera and hand-eye calibration for a robot vision system using a point marker. in Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference. IEEE; 2014
14. Baeg SH, et al. An object recognition system for a smart home environment on the basis of color and texture descriptors. in Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on. IEEE; 2007.
15. Singh P, et al. Domain ontology based efficient image retrieval. in Intelligent Systems and Control (ISCO), 2013 7th International Conference. IEEE; 2013.
16. Chavez-Aragon A, Starostenko O. Image retrieval by ontological description of shapes (IRONS), early results. in Computer and Robot Vision, 2004.

Proceedings. First Canadian Conference. IEEE; 2004.
17. Lemaignan S, et al. ORO, a knowledge management platform for cognitive architectures in robotics. in Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference. IEEE; 2010.
18. Deplanques Y, et al. An ontology for robot modelling and testing; 1996.
19. Bukhari AC, Tusseyeva I, Kim YG. An intelligent real-time multi-vessel collision risk assessment system from VTS view point based on fuzzy inference system. Expert Systems with Applications. 2013; 40(4):1220-1230.
20. Mi ZS, Bukhari AC, Kim YG. An Obstacle recognizing mechanism for autonomous underwater vehicles powered by fuzzy domain ontology and support vector machine. Mathematical Problems in Engineering; 2014.
21. Gómez JB, et al. Mouth gesture and voice command based robot command interface. in Robotics and Automation, 2009. ICRA'09. IEEE International Conference. IEEE; 2009.
22. Prud'hommeaux E, Seaborne A. SPARQL query language for RDF. W3C Recommendation; 2008.
23. Open CV. Open CV documentation. Available:http://docs.opencv.org/ [cited 2015 10/5]
24. Bechhofer S. OWL: Web ontology language, in Encyclopedia of database systems. Springer. 2009;2008-2009.
25. McGuinness DL, Van Harmelen F. OWL web ontology language overview; 2004.
26. Protégé. Protégé. Available:http://protege.stanford.edu/ [cited 2015 1/3]
27. Horrocks I, et al. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission. 2004; 21:79.
28. Parsia B, Sirin E. Pellet: An owl dl reasoner. in Third International Semantic Web Conference-Poster; 2004.
29. Sesame. Sesame. Available:http://rdf4j.org/ [cited 2015 2/4]
30. Horridge M, Bechhofer S. The OWL API: A Java API for OWL ontologies. Semantic web–interoperability. Usability, Applicability. 2011;2(1):11–21. DOI: 10.3233, SW-2011-0025
31. Özdemir B, et al. Performance measures for object detection evaluation. Pattern Recognition Letters. 2010;31(10):1128-1137.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/13329