# A Scalable Method for Robot Object Recognition Using Augmentable Ontology

**Jeahyun Park**, **Juyoung Park**, **Joongmin Choi**
*Dept. of Computer Science and Engineering*
*Hanyang University, Ansan, Korea*
*{jhpark, parkjy, jmchoi}@cse.hanyang.ac.kr*

**Gi Hyun Lim**, **Il Hong Suh**
*Dept. of Information and Communications*
*Hanyang University, Seoul, Korea*
*hmetal@incorl.hanyang.ac.kr, ihseo@hanyang.ac.kr*

*Abstract*— **An intelligent robot must have knowledge acquired and maintained either locally or remotely. When a robot needs to recognize an object, it would make a decision based on the information in the knowledge base. Conventional ways to construct the knowledge models, especially during the object recognition process, are quite rigid and not scalable.**

**In this paper, we propose a new flexible recognition model using the ontology concept. Ontology provides platform-independent representation for data and schema for the relationship among data. By using this, the numerical data collected through the sensory units of the robot is transformed into more semantic and meaningful form, and this semantic information helps to determine what object has been detected. This ontology-based approach contributes to enhance the object recognition performance, especially in an environment where unexpected situations are encountered. We implemented a knowledge management system for intelligent robots, and the results of testing ontology models in this environment are satisfactory.**

## 1 INTRODUCTION

Considering the current technologies for sensors and processors with limited capability, it is not an easy task for a robot to recognize objects around it and determine their identities[4, 6]. In general, robots rely on the visual information when they try to detect and measure the features of the objects for recognition. Cameras mounted on a robot capture the image and the robot translates it into a digital form. The data then must be processed through a number of refining phases and analyzed to be understood by the robot. Each phase and process is determined by a model that the robot has.

Most current models for object recognition are very rigid and hard to be modified and refined once they are established. New types of objects may not be easily added to the model, and moreover, the number of kinds of objects recognizable cannot be extended because the object and its specification to which the model was instantiating have to be fixed and registered in advance[12].

To solve this problem, this paper proposes a flexible and scalable method to recognize objects using the representation of ontology. Ontology is a platform-independent representation of data which is able to be adapted to many domains and applications[2, 11]. In philosophy, ontology means the real nature of entities. The specification of the objects might correspond to the nature[5]. In computer science, ontology used to be referred as the information or knowledge represented declaratively. The knowledge in the ontology is flexible and scalable by this declarativeness. When we have a new fact, we can extend the model with just a simple addition to the ontology. If there is a need to change some data, we only have to modify the knowledge base instead of the model itself. This approach is obviously different from the method that uses the complete metadata, and we try to emulate human recognition of objects.

## 2 ROBOT OBJECT RECOGNITION USING VISUAL INFORMATION

An image captured by robot cameras consists of pixels. Because the image is seen as a set of pixels, the robot is not able to understand what is useful and what is meaningless by just examining these low-level data. In computer vision, the object recognition is informally defined as the procedure that

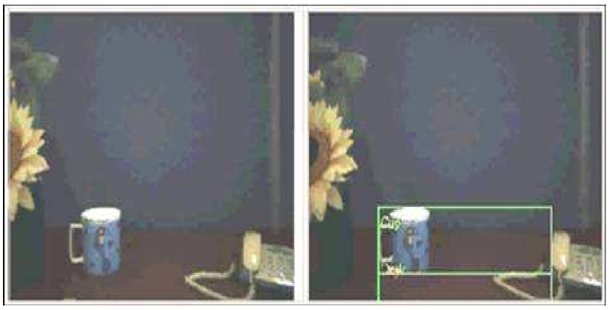

**Figure 1. A segment in an image**

distinguishes objects from the background and determines the identity of the object after characterizing some features in appearance.

There are some portions in an image that we want to extract for recognition and we call each part a *segment*. Recognizer examines each segment independently, and decides what object is in each segment. Of course there may be no object in a segment. The remaining area excluding the segments is regarded as the meaningless background. Fig 1 shows an example of a segment in an image.

Each segment must be analyzed further in a certain way to recognize objects in it. Using the pixel information of the segment, the recognizer analyzes the image statistically for extracting special features of the object. There are some measurement models to figure out prominent features of the segment as explained below.

A. RGB model

RGB model represents each color of the pixel by three colors such as red, green, and blue[3]. Typically, each color is quantized from 0 to 255 degree. A value that represents the entire image segment can be determined by the minimum, maximum or average value of the pixels.

B. HSV model

HSV model represents each color of the pixel by three metrics such as hue, saturation, and intensity[3]. Each metric has a wide range of values of its own, and they are orthogonal to each other. Similar to RGB model, a value that represents the entire image segment can be determined by the minimum, maximum or average value of the pixels.

C. SIFT

SIFT has been developed to find identical objects lied on different images[10]. Even for the same objects, there are possibilities that many images could be obtained in different directions and different sizes. In the SIFT algorithm, a segment has feature vectors, and if two images have a common vector, we can decide that the two images must contain the same object.

**Table 1. Three primitives of the description logic**

| | |
|---|---|
| Class Schema | A class indicates a concept that is treated by robots and stored in the knowledge base. The schema includes information about the hierarchical relationship and property connectivity between the concepts. This relationship will be used for ontology reasoning. |
| Individual Set | A set of entities which are instantiated to a certain class. These entities follow the schema defined for their class. |
| Rule Box | A set of rules related to the class schema and the individual set. Primary axioms declared in the ontology and restrictions are also included. |

In fact, it is not quite appropriate to make a decision about what object is in an image segment using just a single standard. Each measurement has different dimensionality and different ranges of values, so we adopt a novel method to combine these measurements like humans do. This is the reason why we need an abstract semantics that must be employed for the recognition process.

## 3 ONTOLOGY FOR CONCEPTUAL REPRESENTATION

In this paper, we use ontology to represent and share the information which robots accept and interpret, and to extend it easily. The content and meaning of the information in the ontology should be self-describable and independent of specific system characteristics. In this paper, we use the Web Ontology Language (OWL) to construct the object ontology and other kinds of ontologies that the robot needs for the object recognition[13]. OWL is founded by W3C and based on the description logic. Table 1 shows three primitives of the description logic[1]. Naturally, OWL has axioms corresponding to each primitive to build a knowledge base.

First of all, the objects which need to be recognized by the robots should be specified in the ontology. The advantages of the ontology representation include the scalability to extend its content and kinds of the objects, and also the easiness for organizing concepts and their attributes. The description logic provides a framework for establishing the hierarchical relationship between concepts, widely known as the super-class and sub-class hierarchy. Fig. 2 shows a part of the object ontology we made. Ontology concepts are denoted with a circle mark and a tiny triangle represents the hierarchy among the concepts. As you notice in the figure, we


**Figure 2. Class hierarchy**

build the ontology using Prot$\acute{e}g\acute{e}$ which is one of the mostly used ontology development tools[9].

Another merit of ontology is that we can take advantage of special software called the *ontology reasoner*[8, 14]. There are already many intelligent modules to set up and handle the knowledge base written in the description logic. The robots might be connected to the reasoner to use the information gathered interactively.

## 4 RESTRICTIONS AND RULES

To carry out the process of recognizing an object using ontology, both rules and restrictions are to be used. The rules make the lower level, sensory information meaningful by mapping the numerical data to semantic information. This information is more abstract than low-level data and more human-readable. The recognizer is able to do more subjective and trusty interpretation using it. Moreover, the richer inference is available if we have optional contextual information. After the generation of new semantic information, the restrictions of the ontology are applied. These restrictions (or restrictive constructions) are used for classifying each object in a segment to a specific entity.

### 4.1 Rules

Typically, a rule in the description logic has the form as in (1).

$$pred_1(values_1) \wedge \cdots \wedge pred_n(values_n) \rightarrow concept(entity) \quad (1)$$

In (1), each $pred_i(values_i)$ is a sentence. They are connected conjunctively and constitute the condition to be satisfied for firing the rule. As mentioned above, in the ontology, these rules make new information that is more abstract than the lower-level visual sensor data. For example, a rule that determines the color of a segment as red using visual data obtained from the camera mounted on a robot can be written as (2).

$$\begin{aligned} & Hue(?x) \wedge value(?x, ?v) \wedge (?v > ?lb) \wedge \\ & sameAs(?bx, \texttt{RANGE\_OF\_RED}) \wedge low(?bx, ?lb) \wedge \\ & high(?bx, ?ub) \wedge (?v < ?ub) \\ & \longrightarrow Red(?x) \end{aligned} \quad (2)$$

Notice in this rule that RANGE_OF_RED has two attributes, low and high, which indicate the lowest value and the highest value of hue by which the segment color can be classified to red, respectively. If the hue value of the segment lies between the two bounds, the segment will have the *hasHue* property, and its value will be an instance of the class *Red*. The reason why RANGE_OF_RED is defined indirectly is for flexibility. If we need to change the range of the hue value for classifying to *Red*, we just modify two numeric values in the ontology. For robot object recognition, we also use some other features such as saturation, intensity, shape, and size, and these features are similarly defined by their own semantic range descriptions.

Unfortunately, the OWL language basically do not support features for describing rules. Hence, in order to represent rules appropriately, we use the SWRL language[7] that is based on RuleML and expresses rules within the OWL framework. Fig. 3 shows a list of SWRL rules for object recognition that are built by using the Prot$\acute{e}g\acute{e}$ SWRL plug-in.

### 4.2 Restrictions

A restriction in the description logic are used to construct a concept node which can be regarded as a set of individuals. For instance,

$$\exists hasHue.Red \quad (3)$$

denotes a restriction for a set of individuals which must have an attribute *hasHue* and its value must be an instance of the concept *Red*. Without loss of generality, we can say that this notation is a way of defining a new concept from other concepts. By using this restriction, we can build any complex concept as in (4).

$$\begin{aligned} Apple \sqsubseteq Object \sqcap \; & \exists hasHue.Red \sqcap \\ & \exists hasSaturation.Strong \sqcap \\ & \exists hasIntensity.Medium \sqcap \\ & \exists hasShape.Circle \sqcap \\ & \exists hasSize.Small \end{aligned} \quad (4)$$

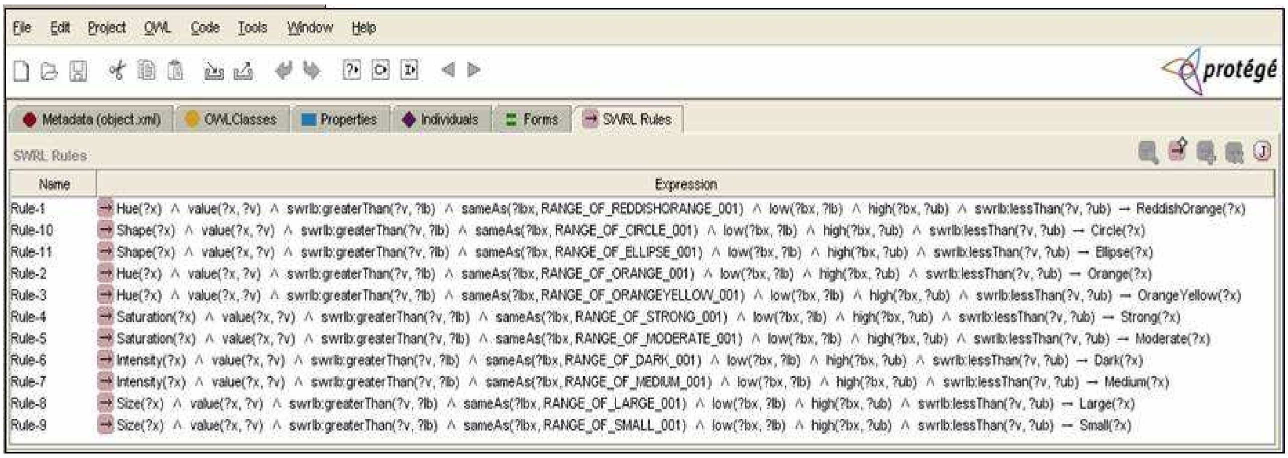


**Figure 3. Rules in SWRL**

**Table 2. Restrictions for 3 different objects**

| | |
|---|---|
| Apple | $\exists\ hasHue.Red$ |
| | $\exists\ hasIntensity.Medium$ |
| | $\exists\ hasSaturation.Strong$ |
| | $\exists\ hasShape.Circle$ |
| | $\exists\ hasSize.Small$ |
| Bread | $\exists\ hasHue.OrangeYellow$ |
| | $\exists\ hasIntensity.Medium$ |
| | $\exists\ hasSaturation.Strong$ |
| | $\exists\ hasShape.Ellipse$ |
| | $\exists\ hasSize.Large$ |
| Steak | $\exists\ hasHue.Orange$ |
| | $\exists\ hasIntensity.Dark$ |
| | $\exists\ hasSaturation.Moderate$ |
| | $\exists\ hasShape.Ellipse$ |
| | $\exists\ hasSize.Small$ |

This restriction indicates that an object instance with red hue, strong saturation, medium brightness, circle shape, and small size can be regarded as an apple. The restriction is a practical description of the appearance of an apple, and at the same time we can use this restriction to make some instances that are not known before to be classified to the correct concept. If you're using reasoners, the restriction gets applied automatically and new instances in the apple concept will be instantiated. Restrictions can be formally described in OWL, and Fig. 4 is an example of the apple restriction. Naturally, all sub-concepts of *Object* have their own restrictions, and Table 2 shows the restrictions for *Apple*, *Bread*, and *Steak* in the ontology.

Notice that we only use the existential quantifiers in specifying restrictions. This implies that some instances might be classified into multiple concepts simultaneously, since the condition for precise classification is satisfied if there is just one attribute which has a proper value. This problem of multiple classification can be partially solved by considering in-

```
<owl:Class rdf:ID="Apple">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Object"/>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#Red"/>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasHue"/>
            </owl:onProperty>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasIntensity"/>
            </owl:onProperty>
            <owl:someValuesFrom rdf:resource="#Medium"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="Strong"/>
            </owl:someValuesFrom>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasSaturation"/>
            </owl:onProperty>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="Circle"/>
            </owl:someValuesFrom>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasShape"/>
            </owl:onProperty>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#Small"/>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasSize"/>
            </owl:onProperty>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
```

**Figure 4. Restrictions for *Apple* in OWL**

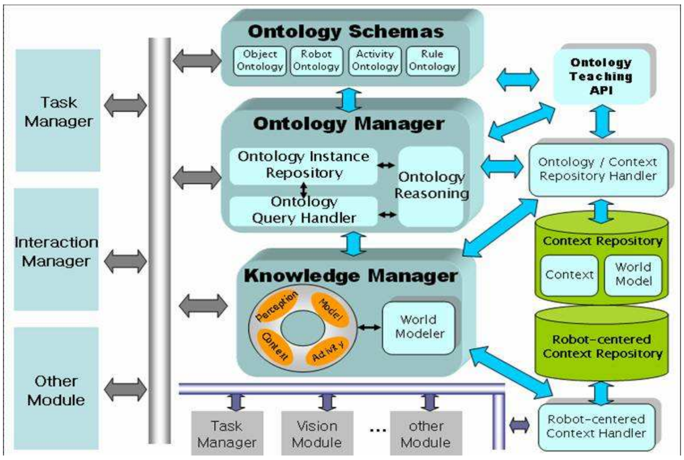


**Figure 5. A knowledge management system for intelligent robots**

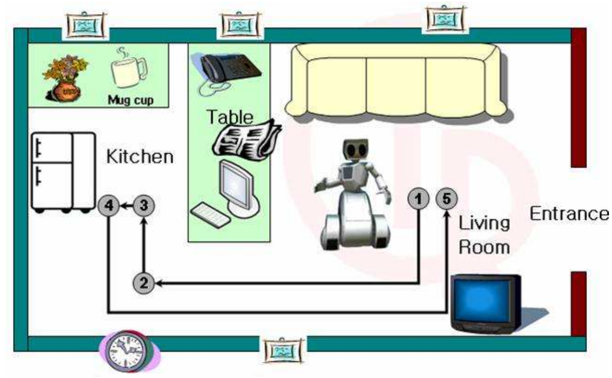


**Figure 6. Test environment for robot task of finding a cup**

**Table 3. APIs for ontology handling**

| SIGNATURE | RETURN |
|---|---|
| createOntologyInstance(classURI) | instanceID |
| getOntologyInstances(classURI) | instanceID[] |
| getProperties(resourceURI, type) | propertyID[] |
| getPropertyValues(instanceURI, propertyURI) | value[] |
| setPropertyValue(instanceURI, propertyURI, value) | None |
| retractOntology(uri) | None |
| retractPropertyValue(instanceURI, propertyURI, value) | None |
| update(subject, predicate, object) | None |

formation from additional visual components that can reduce the number of possible classifications according to their own restrictions.

## 5 A KNOWLEDGE MANAGEMENT SYSTEM FOR INTELLIGENT ROBOTS

The object ontology described in previous sections is exploited to implement a knowledge management system for intelligent robots. The overall architecture of this system is shown in Fig. 5. Ontology manipulation is processed mainly by the ontology manager that is controlled by the knowledge manager. To communicate with the knowledge manager for handling ontology, we provide APIs as shown in Table 3.

The overall recognition process is triggered when new visual information is entered via robot cameras under a plan established by the upper scheduler. This process is repeated every time the robot gets new visual information. The reason why the recognition is initiated and repeated with the external visual information instead of managing explicit submechanism for recognition is that, in most cases, the input images are not unique even when the images contain the same object, and the robot should react immediately in most tasks. Once the inputs are accepted, a time tag is also assigned to each snapshot. Eventually, these time tags comprise the contextual information that can be exploited later when the robot faces a similar situation.

The ontology-based knowledge management system is implemented and tested with a mobile robot for a task of finding a cup inside home. The test environment for this robot task is shown in Fig. 6.

In this scenario, the robot is ordered to find a cup in the kitchen. At first, the robot tries to search a region with the highest probability of containing a cup, and then retrieves a path to the region from the place the robot is standing. When the robot moves following the path, the cameras accept images and the recognizer translates these data into ontology and infers the identities of the objects using the rules and restrictions. Robot testing snapshots are shown in Fig. 7 and Fig. 8. Especially, the ontologies generated during the execution are shown by the right side box in Fig. 8.

The primary goal of this robot is to unveil the object in the image segment to find a cup. However, even if the object recognized is not a cup, the information about the object is useful and stored for later use, since it contributes to avoid unnecessary path retrieval when the same object is to be searched again.

## 6 CONCLUSIONS

This paper have proposed a scalable recognition model using the ontology concept. Ontology provides platform-independent representation for semantic information for low-level visual data. Ontology-based knowledge management approach contributes to enhance the object recognition performance, especially in an environment where unexpected situations are encountered. In future, additional special-purpose intelligent components, such as the reasoner, could make the system behave more efficiently, and the pervasive environment could make deductive knowledge more valuable.


**Figure 7. Test-bed robot for testing**

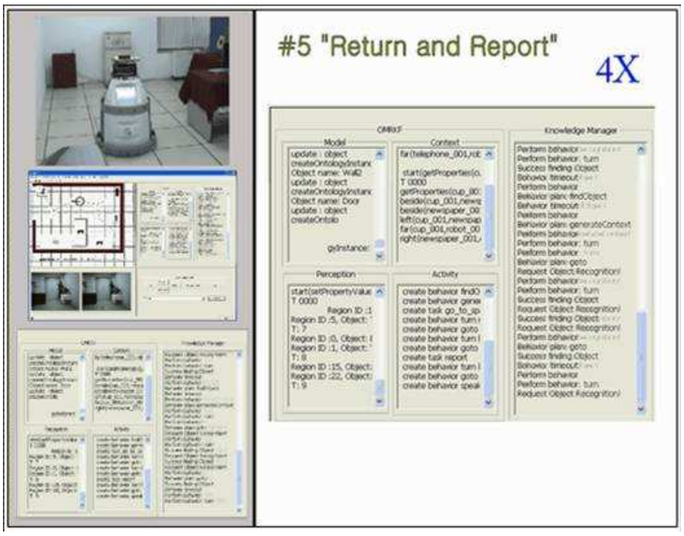


**Figure 8. A snapshot in robot testing**

## ACKNOWLEDGEMENT

This work was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by Korea Ministry of Commerce, Industry and Energy.

## References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, 2003.

[2] D. Fensel, M. Brodie, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd ed., Springer Verlag, 2003.

[3] J. Foley, A. van Dam, S. Feiner, J. Hughes, R. Philips, *Introduction to Computer Graphics*, Addison-Wesley, pp. 410–418, 1997.

[4] R. Gonzalez, R. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall, 2002.

[5] T. Gruber. "A Translation Approach to Portable Pntologies", *Knowledge Acquisition*, Vol.5, No.2, pp.199-220, 1993.

[6] A. Hanson, E. Riseman, *Computer Vision Systems*, Academic Press, New York, 1978.

[7] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", http://www.w3.org/Submission/SWRL, 2004.

[8] KAON2, http://kaon2.semanticweb.org

[9] H. Knublauch, R. Fergerson, N. Noy, M. Musen, "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications", *Proceedings of 3rd International Semantic Web Conference*, 2004.

[10] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Vol.60, No.2, pp.91-100, 2004.

[11] A. Maedche, *Ontology Learning for the Semantic Web*, Springer Verlag, 2006.

[12] T. Matsuyama, V. Hwang, *SIGMA: A Knowledge-based Aerial Image Understanding System*, Springer Verlag, 2004.

[13] D. McGuinness, F. van Harmelen, "OWL Web Ontology Language: Overview", http://www.w3.org/TR/owl-features, 2004.

[14] B. Parsia, E. Sirin, "Pellet: An OWL-DL Reasoner", *Proceedings of 3rd International Semantic Web Conference*, 2004.