Intel Serv Robotics (2010) 3:115–123
DOI 10.1007/s11370-010-0063-6

ORIGINAL RESEARCH PAPER

Robust robot knowledge instantiation for intelligent service robots

Gi Hyun Lim · Il Hong Suh

Received: 13 January 2010 / Accepted: 16 February 2010 / Published online: 3 March 2010

Abstract Robot knowledge is considered to endow service robots with intelligence. In the real environments, robot knowledge needs to represent dynamically changing world. Despite its advantages for semantic knowledge of service robots, robot knowledge may be instantiated and updated by using imperfect sensing data, such as misidentification of object recognition. In case of using commercially available visual recognition system, incorrect knowledge instances are created and changed frequently due to object misidentification and/or recognition failures. In this work, a robust semantic knowledge handling method under imperfect object recognition is proposed to instantiate and update robot knowledge with logical inference by estimating confidence of the object recognition results. The following properties may be applied to determine misidentifications in logical inference: temporal reasoning to represent relationships between time intervals, statistical reasoning with confidence of object recognition results. To show validity of our proposed method, experimental results are illustrated, where commercial visual recognition system is employed.

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.

G. H. Lim
Department of Electronics and Computer Engineering,
Hanyang University, Seoul, Korea
e-mail: hmetal@hanyang.ac.kr

I. H. Suh (✉)
Division of Computer Science and Engineering,
College of Engineering, Hanyang University, Seoul, Korea
e-mail: ihsuh@hanyang.ac.kr

Keywords Robot knowledge · Robust knowledge instantiation · Robust knowledge update · Confidence interval · Temporal reasoning · Logical reasoning · Statistical inference · Ontology

1 Introduction

Robot knowledge is considered to endow service robots with intelligence [1,2]. Service robots need to reason about their plan and/or query for more information [3] to complete their mission. In the real environments, robot knowledge needs to represent dynamically changing world. For example, before a robot prepares to enter a room, it is required to know whether the door is open or closed. If the door is closed, the robot needs to plan additional actions, such as finding door knob and opening the door. In this way, sensing data are instantiated as facts, and additional facts are inferred from those facts.

Instantiation and update of semantic knowledge are based on perfect facts [4] without recognition failure or misidentification. Useful logical inference requires robust robot knowledge even under conditions of imperfect symbol grounding [5]. Despite the need for service robots to have semantic knowledge, robot knowledge may be instantiated and updated by using imperfect sensing data, such as misidentification of object recognition. Even if commercially available visual recognition systems are used [6], incorrect knowledge instances are created and changed frequently due to object misidentification and/or recognition failures because of mismatches [7].

This paper introduces a robust robot knowledge instantiation (RoKI) method for use under conditions of imperfect sensing data; the method can be applied to determine false positives and false negatives and to estimate confidence of object recognition results for logical rigidness. The method involves the attachment of vision sensors to the robot, which can then move around to recognize and localize objects in an environment. At times certain objects may be correctly identified, whereas at other times the same objects may be incorrectly identified or completely overlooked. At the time of recognition, it is not possible to ensure correct object recognition results, so the recognized object cannot be registered at that moment. It is first necessary to verify whether the recognized objects and/or created spatial instances are correct. The method addresses this by temporarily storing recognized objects in two buffers, and these objects generate intervals until they are verified as correct or incorrect. The following properties may be applied to determine misidentifications in logical inference: temporal reasoning to check the validity of relationships between time intervals, statistical reasoning to determine the confidence level of object recognition.

This paper is organized as follows: Sect. 2 discusses previous researches; Sect. 3 outlines the proposed robust RoKI method; Sect. 4 explains how temporal confidence reasoning can be used to determine false positive and false negative results, respectively; Sects. 5 and 6 discuss the experimental results that show the validity of the method; and Sect. 7 presents concluding remarks.

2 Related works

Failures caused by misidentified sensing data are a common occurrence, but if at all possible service robots should minimize or avoid the effects of failure. Robots need to be able to deal with failure resulting from four kinds of data: incomplete and uncertain temporal information, the projecting of events before they occur, the effects of actions as long as they are relevant (even if the available time does not permit), and the reasoning process [8].

To address failures, researchers have come up with various ways including probabilistic approaches [9,10], heuristic methods [11,12], and rule-based approaches [13–16]. Among them, Thrun has used probabilistic approaches to guide robots toward their goal position with uncertain and noisy sensor data [9], and probabilistic boolean networks were introduced to face uncertainty [10]. As heuristic methods, Jahanian et al. [11] suggested randomization-based fuzzy rule methods to control the false positive rate, and Lim et al. [12] proposed a method that can be used to suggest alternative actions to make up for the incomplete and uncertain perceptual information. Kyoshgoftaar [13] applied rule-based method based on boolean rules to detect noisy data. Moreover, rule-based approaches have been proposed for coping with any uncertainties or vagueness of misidentification in object recognition [14]. On the other hand, some researchers have attempted to manage uncertainties in logic programming [15]. Ding et al. [16] proposed approaches for covering uncertainties in ontology and attempted to extend these to OWL ontology [17] using Bayesian networks. Probabilistic approaches and heuristic methods are likely useful approaches for managing uncertainty, but they have their own weaknesses, including a scale problem that occurs when additional elements are added because the probability represents propositional knowledge of the sentence itself not predicate sentence [18]. Rule-based systems have the advantages of locality, detachment, and truth-functionality and therefore can be used as expert systems to help humans make decisions about a specific problem domain with rules gathered from expert knowledge. In our system, rule-based method is considered to determine misidentification.

3 A robust robot knowledge instantiation (RoKI) method

Usually, recognition systems yield false negative results more frequently than false positive results because of illumination changes, motion blur, occlusion, and so on (see Fig. 1), so many researchers have concentrated on determining invariant features about environment changes that may eliminate false negative results. False positive results are not a serious impediment to visual recognition but can be problematic for a knowledge-based system in dynamic environments. Insufficient facts due to false negative results can be corrected by additional true positive results, but erroneous facts due to false positive results will result in false consequences for reasoning; this generates a vicious cycle, and errors are difficult to correct even with additional true negative results. Figure 2 presents knowledge instances for a false positive case: a `pot` that is actually located on a table in the kitchen but is recognized as being to the left of the `refrigerator`, so the location of `pot` may be updated and spatial relations, such as `pot` is to the left of `refrigerator` may be inferred. At that time, if a query is made about the `pot`'s location, then the knowledge system will provide an answer based on what it knows, e.g., "The `pot` is to the left of the `refrigerator`, not on the table." This answer will prevent the robot from finding the `pot`.

To address the failure of knowledge instantiation under conditions of imperfect object recognition, we developed a dependable semantic knowledge instantiation method to ensure logical rigidness of robot knowledge instances.

Figure 3 is a system diagram of the proposed method. In the system, rules are designed to confirm the authenticity of object recognition to ensure dependable knowledge instantiation. Two reasoning mechanisms are considered in a generic manner for logical inference: temporal reasoning and statistical confidence interval (CI). In addition, ontological representation is used for domain-specific knowledge.

Fig. 1 Examples of misidentification

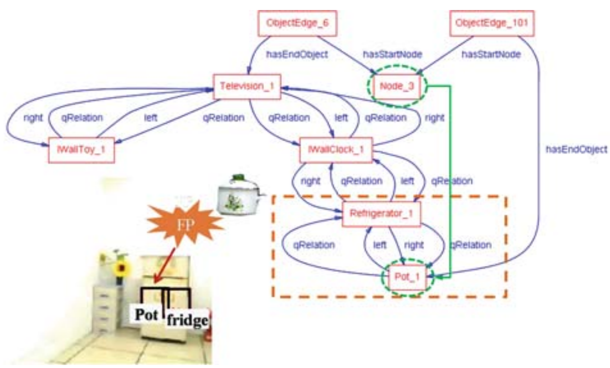

Fig. 2 Knowledge instances of misidentification about a `Pot`

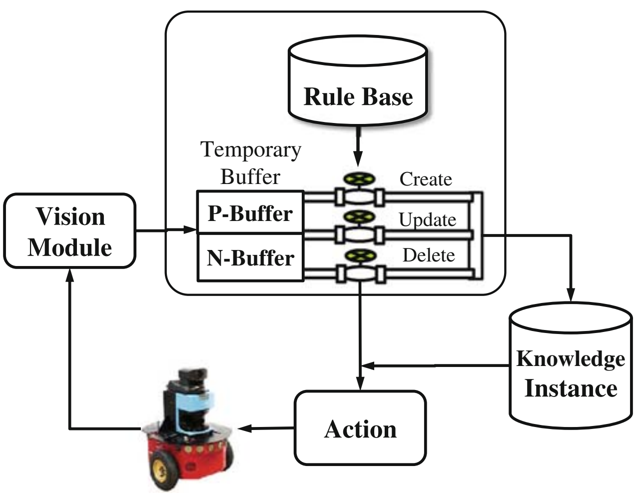

Fig. 3 System diagram of the proposed method

When objects are recognized, the results are stored to buffers, such as p-buffer or n-buffer. Intervals, such as *is*-interval or *has-to-be*-interval, are generated from data in the buffers for each object. Through the application of rules for true positive, true negative, false positive, and false negative to the generated intervals, the recognition results are verified as true or false and instantiated into the robot knowledge instance database.

Reasoning mechanisms use four properties to build rules in the object detection method: temporal reasoning to check the validity of relationships between time intervals [19], statistical reasoning to determine the confidence level of the object recognition [20], ontological reasoning to check if a detected object satisfies object properties [21,22]. Temporal confidence reasoning in particular is designed as a generic component, whereas ontological and spatial reasoning represent domain-specific knowledge with their ontological instances.

4 Temporal confidence reasoning (TCR)

4.1 Temporary buffers and intervals

It is difficult to verify recognition results at the time of recognition, so two buffers—a positive buffer (p-buffer) and negative buffer (n-buffer)—were designed to store object recognition results until they could be verified as correct or incorrect. Some objects which are recognized before they are instantiated are stored in the p-buffer. Positive object recognition results are compared to the ones in the instance database, in which object instances are stored after they are verified as true, whereas negative recognition results (which were supposed to have been recognized according to the robot's view) are stored in the n-buffer. To confirm whether data in the buffers are true or not, intervals are measured between instances at which the same object is or should be recognized. There are two types of intervals: *is*-interval and *has-to-be*-interval. The *is*-interval represents positive recognition results (1), and the *has-to-be*-interval represents negative recognition results (0). Intervals for each object are generated in the two buffers. Interval sizes are determined by interval counter.

If any *is*-interval in the p-buffer is satisfied, the recognized results are considered to be true positive. When the object is not yet registered, the objects are instantiated; otherwise, they are updated. With regard to *has-to-be-intervals*, if an object is from the p-buffer, other *is-intervals* are considered to be false positives, so the object is not applied to the instance database. In contrast, if the object is from the n-buffer, it is considered to be excluded from the registered position, and the registered position is updated to "unknown."

4.2 Confidence interval (CI)

Confidence of recognition is determined by interval-counter(γ) from the recognition rate for each object and the recognition rate for each object is obtained statistically. In this method, recognition rates for objects are assumed to be given by experts before rules are applied.

An interval-counter for each object is defined following the *confidence law of inertia*, whereby a knowledge instance is assumed to persist unless there is confidence to believe otherwise. If the recognition rate of object `A` is x_A, (1-x_A) is the probability that the recognition data for `A` can be false. From that, $(1\text{-}x_A)^{\gamma_A}$ can be calculated to define probability when the values of γ_A consecutive data are all false. If the result of $(1\text{-}x_A)^{\gamma_A}$ is less than 5% (0.05), then it can be said that the data are 95% CI (1.96σ, $P = 0.05$) of the confidence level. For example, if the recognition rate of object `A` is 80%, the recognition failure rate of object `A` might be 20% (0.2). Two consecutive recognition failure rates are 4% (0.04) and 0.04 which are beyond the 95% CI($P = 0.05$), so the interval-counter of object `A` is 2. The interval-counter can be represented as follows:

$$\gamma_A = \min\{\gamma \in I | (1 - x_A)^{\gamma} \leq P\} \tag{1}$$

4.3 Temporal and statistical reasoning

In the two buffers, temporal reasoning is used to represent temporal relations between intervals. This kind of temporal relation was first proposed by Allen [19] and represents temporal relations using before, after, meets, met-by, overlaps, overlapped-by, and so on. Table 1 lists the rules of temporal reasoning to show the end point relations between two intervals. In the table, obj_1 and obj_2 are object instances, intervals a_l, a_m and a_n include start point a^s and end point a^e. If two intervals meet or overlap, then they are merged into one interval. The merged interval begins at the start point of the former and ends at the end point of the latter. Temporal confidence reasoning is based on the assumption that recognized objects cannot go away and come back within a single time interval.

4.4 Ontology representation

Whereas hand ontology [17] schema represent generic knowledge, ontology instances can represent domain-specific knowledge that expresses a recognized environment. Spatial relations among objects, such as left, right, above, and so on, are generated from the localized position of objects using spatial reasoning when object instances are created. Spatial relations enable a robot to achieve complex tasks by supporting spatial contexts in the environment. Table 2 lists axioms and their spatial relation properties.

When an object's position is estimated, RoKI registers the object and generates a semantic map, which is composed of a `Node`, `NodeEdge`, and `ObjectEdge`. Each `Node` has an x and a y position. `Nodes` are connected with a `NodeEdge` and construct a `TopologicalMap`. A `NodeEdge` means that the robot can move from one `Node` to the next `Node`. `Nodes` and `Objects` are connected with an `ObjectEdge`, which represents whether an `Object` can be recognized on a `Node`. Figure 4 represents a semantic map consisting of five `Nodes`. Each `Node` is connected to the next with a `NodeEdge`. While the robot moved around,

Table 1 Rules of temporal confidence reasoning (TCR)

Temporal Relation	Example	End Point Relations
if $obj_1 = obj_2$ **and** $a_l^{obj_1}$ **meets** $a_m^{obj_2} \Rightarrow a_n^{obj_1}$	a_l a_m $\Downarrow a_n$	
if $obj_1 = obj_2$ **and** $a_l^{obj_2}$ **met_by** $a_m^{obj_1} \Rightarrow a_n^{obj_1}$		$a_l^s < a_l^e = a_m^s < a_m^e \Rightarrow$ $a_l^s = a_n^s < a_m^e = a_n^e$
if $obj_1 = obj_2$ **and** $a_l^{obj_1}$ **overlaps** $a_m^{obj_2} \Rightarrow a_n^{obj_1}$	a_l a_m $\Downarrow a_n$	
if $obj_1 = obj_2$ **and** $a_l^{obj_2}$ **overlapped_by** $a_m^{obj_1} \Rightarrow a_n^{obj_1}$		$a_l^s < a_m^s < a_l^e < a_m^e \Rightarrow$ $a_l^s = a_n^s < a_m^e = a_n^e$

Table 2 Spatial relation properties

Axiom	Spatial relation property
Transitive property	$right^+ \in \mathbf{R}_+$
	$left^+ \in \mathbf{R}_+$
Inverse property	$right \equiv left^-$

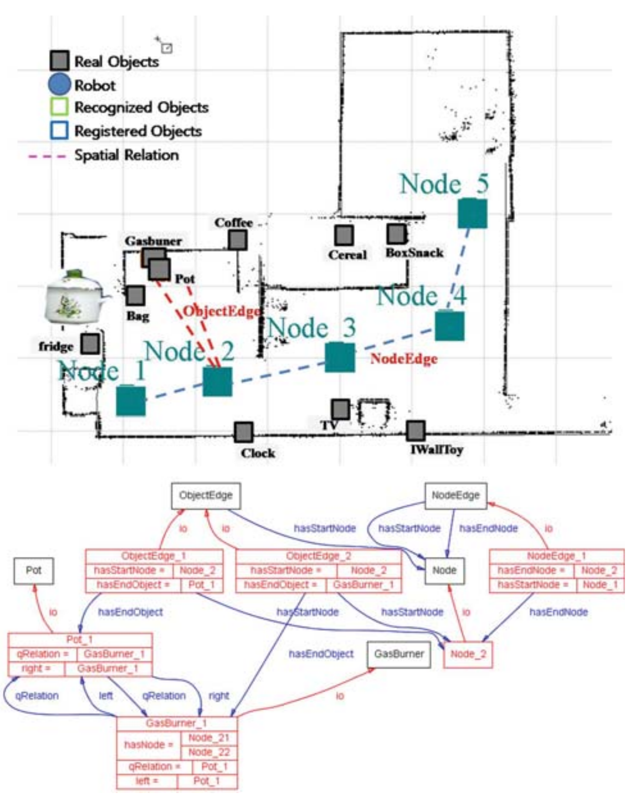

Fig. 4 Example of a semantic map composed of `Nodes`, `NodeEdges`, and `ObjectEdges`

it recognized a `GasBurner` and a `Pot` on `Node_2`. Therefore, the instances of `GasBurner` and `Pot` are linked to `Node_2` with two instances of `ObjectEdge`. If a request is made about `Pot`, the robot will move to `Node_2` to approach the `Pot` instance.

4.5 Applications of TCR rules

Figure 5 shows how our proposed TCR rules work. When object `A` is recognized, a instance for an *is*-interval of object `A` is created. The *is*-interval is ended if the object is not recognized or the interval satisfies the proposed rules. This process generates intervals of object `A`, as shown in Fig. 5. If object `A` is recognized, it is denoted by ‘1’ in the buffer, otherwise it is denoted by ‘0’.

Figure 5a illustrates the determination of false negative and true positive instances using statistical and temporal reasoning rules. The following are used to verify whether or not the object recognition results are confident. Let γ of object `A` is 3. First, three intervals are generated for object `A` (two *is*-intervals and one *has-to-be*-interval) using the process described previously. Whenever an event occurs, the proposed method is applied to verify the authenticity of object recognition. Second, because the interval-counter of *is*-interval a_3^+ is reached at γ of object `A`, the interval can be said to lie within a 95% CI. Thus, by applying the statistical reasoning rules, *is*-interval a_3^+ can be judged as a true positive instance. From this result, *has-to-be*-interval a_3^+ is determined to be a false negative instance and is therefore corrected to be true. Finally, object `A` is considered to have been there the entire time from a_1 to a_3, and the intervals can be merged to a_m^+.

From this sequence, we can determine false positive and true negative instances as same manner of determination of false negative and true positive instances, as shown in Fig. 5b.

When an object instance of `A` is registered, if other objects are also considered to be true positive instances and to have a temporal relation of *overlapped* with object `A`, then spatial relations among the objects might be inferred. For instance, Fig. 5c presents a set of spatial relations between object `A` and object `B`. When the *is*-interval of object `B` is considered to be true, the temporal relation between a_m^+ and b_m^+ is considered to be an *overlap*. Then the spatial relation between them can be reasoned and set using spatial reasoning. Every object instances and their spatial relations can be registered into the instance database.

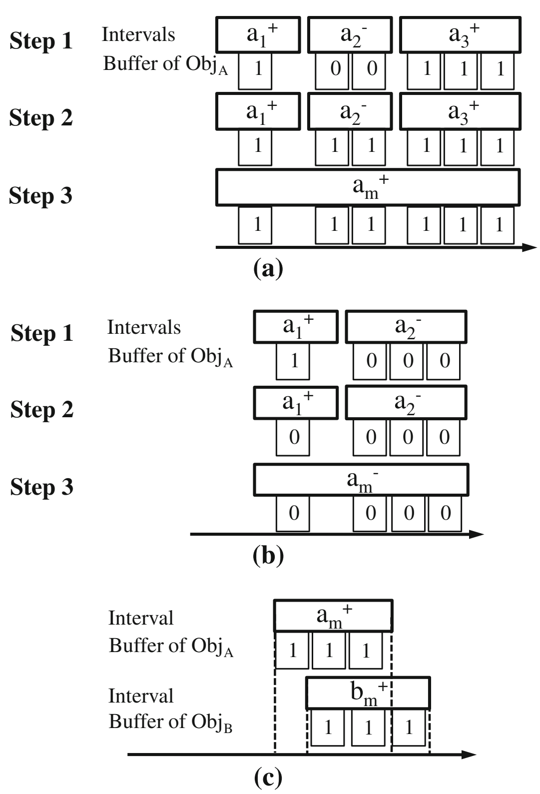

Fig. 5 Mechanism of the proposed method: **a** determination of true positive and false negative instances; **b** determination of a false positive instance; **c** set of spatial relations, in which ‘1’, ‘0’, ‘+’, and ‘−’ denote recognized, unrecognized, *is*-interval and *has-to-be*-interval, respectively

5 Experiments

Figure 6 presents the experimental environment, which was composed of a kitchen and a living room. As much as 11 objects were distributed throughout the environment, as shown by the yellow boxes in Fig. 6. Table 3 lists the recognition rates for each object. The recognition rates were obtained empirically. The robot moved along with the nodes and looked around at every node. During its exploration, the robot took pictures with a single web camera attached on top of robot and recognized each object using an Evolution Robotics Software Platform (ERSP) vision [6] module that returned recognized x and y positions for the image and the distance from the camera to the object, and the visual recognition system interval was about 330 ms. The environment included only one object for each object model.

5.1 Scenario 1: Initial instantiation

The robot first moved from `node 1` to `node 5` and back from `node 5` to `node 1` without any idea about how the objects were distributed. It used the proposed method to generate object instances. Figure 7a shows how the `Television` instance was registered the interval-counter of which is 3. In frame #1, a `Television` is recognized, and an *is*-interval for `Television` was started. The recognized data of `Television` were set into the *p*-buffer. Next, `Television` was recognized two times continuously from #2 to #3. When `Television` was recognized in #3, the

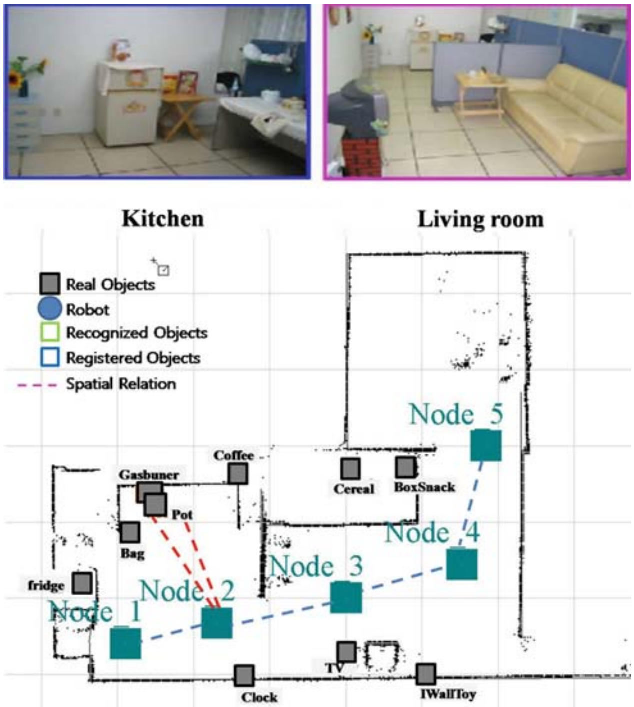

Fig. 6 Experimental environment composed of a `Kitchen` and a `Living room`

Table 3 Recognition rate and γ

γ	Object	Recognition rate
2	WallToy	0.85
2	Cereal	0.80
2	Coffee	0.79
3	BoxSnack	0.76
3	Refrigerator	0.74
3	Bag	0.68
3	Television	0.66
4	GasBurner	0.56
4	Extinguisher	0.54
5	WallClock	0.51
5	Pot	0.51

interval-counter for the *is*-interval of `Television` reached the *confidence level* calculated from the recognition rate of `Television`. By applying the statistical reasoning rules, the method registered `Television` into the instance database.

5.2 Scenario 2: Inference of spatial relations

Figure 7b illustrates a case of set spatial relation between `Cereal` instance and `BoxSnack` instance. Using the statistical reasoning rules, the method registered an instance of

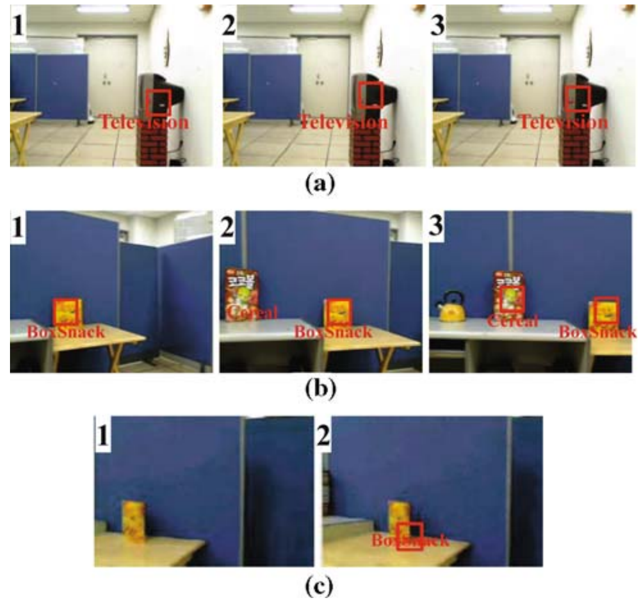

Fig. 7 Sample applications of the proposed method: **a** instantiation of `Television`; **b** inference of a spatial relation between a `Cereal` box and a `BoxSnack`; **c** determining a false negative

`BoxSnack` in frame #2. From frame #2, `Cereal` was recognized and the *is*-interval of `Cereal` was started. At frame #3, `Cereal` was considered to be a true positive according to the statistical reasoning rules, and it was registered into the instance database. When this occurred, the system reasoned that the *is-intervals* of `BoxSnack` and `Cereal overlapped`. As `BoxSnack` and `Cereal` instances were registered as having a temporal relation that `'overlapped'`. the spatial relation between them was set as #3.

5.3 Scenario 3: Determining false negative instances

Figure 7c presents an example of determination of false negative instance of `BoxSnack`. The `BoxSnack` instance was supposed to be recognized at #1 in Fig. 7c, but it was not recognized. Therefore, the unrecognized sign were set into the *n*-buffer, and a *has-to-be*-interval of `BoxSnack` began. In #2, `BoxSnack` instance was recognized, and an *is*-interval was started. When rules were applied to the intervals, the `BoxSnack` instance was recognized at the registered position. Thus, it was considered to be true. Because the `BoxSnack` instance was considered to be at its registered position, the *has-to-be*-interval was considered to be a false negative instance and was merged into the *is*-interval.

5.4 Scenario 4: Determining false positive instances

During the robot's exploration, `Pot` was recognized before `Refrigerator`. Without using the proposed method, the

Pot instance was registered, as shown in Fig. 8a. This instance of Pot, which was registered because of misidentification, could make the set of robot knowledge inconsistent.

However, we found that the proposed method successfully determine this sort of misidentification. #1 in Fig. 9, an *is*-interval of Pot was generated. In the next frame, Pot was not recognized, and a *has-to-be*-interval of Pot was started. From that time, Pot was not recognized consecutively by #6. On #6, the interval-counter of the *has-to-be*-interval of Pot reached γ: 5, calculated from the recognition rate of Pot. This meant that the *has-to-be*-interval was within the 95% confidence level. So the *is*-interval was considered to be a false positive instance, and it was not registered into the instance database as shown in Fig. 8b.

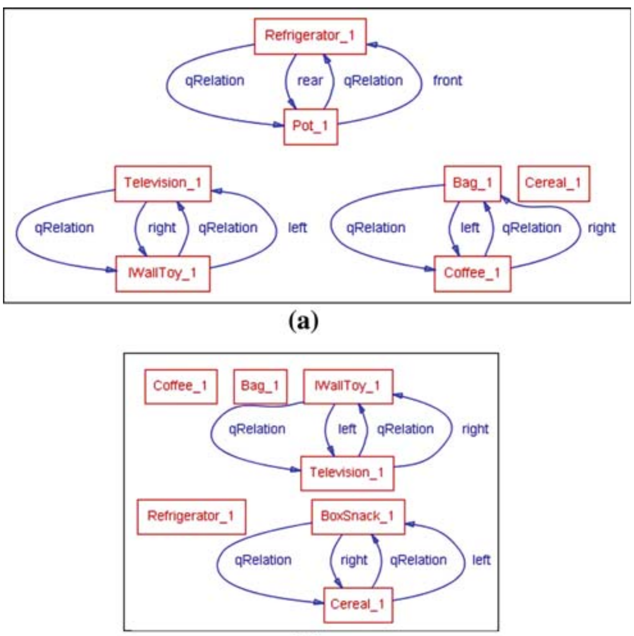

Fig. 8 Knowledge instances **a** without the proposed method, where Pot was registered even when it was a false positive; **b** with the proposed method, where Pot was determined as a false positive

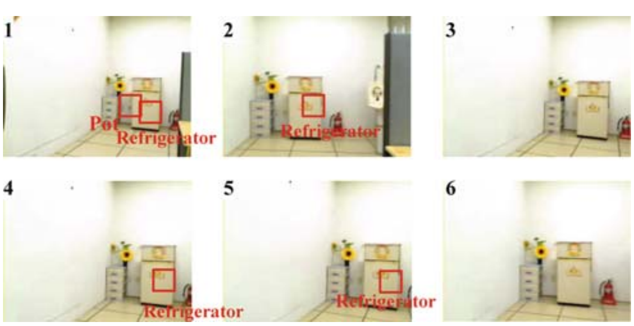

Fig. 9 Example of determination of false positive instance: Pot

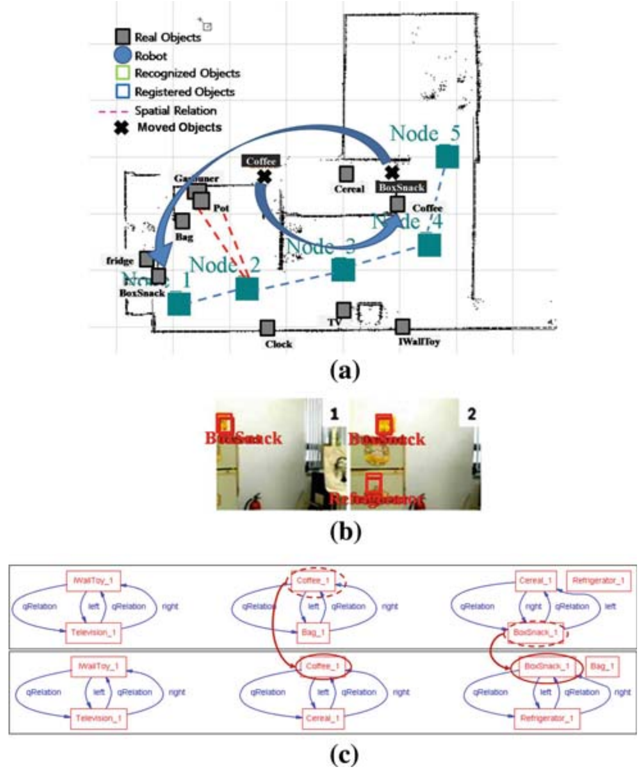

Fig. 10 Example of updating moved instances: **a** experimental environment in which Coffee and BoxSnack were moved; **b** updating the BoxSnack object instance; **c** knowledge instances updated using the proposed method

5.5 Scenario 5: Updating moved instances

To determine whether the proposed method could handle moved objects, we moved Coffee and BoxSnack, as shown in Fig. 10a. The moved instances, even the spatial relations, were updated correctly using the proposed method. For example, the BoxSnack instance was recognized from #1 to #2, as shown in Fig. 10b. By using statistical and temporal reasoning rules, the method updated the registered position of BoxSnack instance in which the updates rules are based on the assumption that every instances can be identified. When it was updated, the Refrigerator instance was also considered to be a true positive, having an 'overlapped' temporal relation with the BoxSnack instance. Therefore, the spatial relation was also updated. The updated instance database is presented in Fig. 10c.

6 Discussion

Table 4 lists the results of RoKI for true positives, false positives, and false negatives for each object, including the number of cases without RoKI (wo), the number of cases with RoKI (w), and the improvement rate. Three objects

Table 4 Recognition results

		TP			FP			FN		
γ	Obj	wo	w	imp	wo	w	imp	wo	w	imp
2	WT	73	109	49.3	2	0	100	42	6	85.7
2	Cer	27	30	11.1	4	0	100	6	3	50.0
2	Cof	18	20	11.1	6	0	100	10	8	20.0
3	BS	20	28	40.0	9	0	100	9	1	88.9
3	Ref	70	85	21.4	4	0	100	29	14	51.7
3	Bag	10	18	80.0	4	0	100	56	48	14.3
3	TV	49	71	−44.9	3	0	100	56	17	69.6
4	GB	40	38	−5.0	3	0	100	17	15	11.8
4	Ext	14	13	−7.1	4	0	100	7	7	0
5	WC	4	1	−75.0	1	0	100	7	10	−42.9
5	Pot	13	6	−53.8	2	0	100	27	34	−25.9

had interval-counter 2: `WallToy` (WT), `Cereal` (Cer), and `Coffee` (Cof); four objects had interval-counter 3: `BoxSnack` (BS), `Refrigerator` (Ref), `Bag`, and `Television` (TV); two objects had interval-counter 4: `GasBurner` (GB) and `Extinguisher` (Ext); and two objects had interval-counter 5: `WallClock` (WC) and `Pot`.

From Table 4, it is observed that all false positives from recognition results were successfully removed, but false negatives were not completely recovered. This can be explained as follows; in the case of vision sensors, object misidentification rates are usually governed by false negatives. In our experiment, for example, the recognition rate of `Television` by Evolution Robotics Vision system was measured as 66%. This implies that misidentification rate could be only 34%. Among 34%, it was observed that false positive rate was 5%, and false negative rate was 29%. Thus, to detect false positives for television, interval-counter for 95% recognition rate will be sufficient according to our proposed (1) in Sect. 4. But, in actual application, interval-counter for 66% recognition rate was applied, which made it to use somewhat excessive interval-counter. Use of excessive interval-counter gets clear removal of false positives but delayed decisions. On the other hand, false negatives were improved as expected since its interval-counter for 66% was used instead for 71%.

With regard to knowledge instances after initial registration trial, 7 objects were registered among 11 objects to be registered: `Coffee`, `Bag`, `WallToy`, `Television`, `Refrigerator`, `BoxSnack`, and `Cereal`. All these successfully registered items had a rather high recognition rate which produces interval-counter values less than 4. Unregistered four objects had interval-counter values greater than or equal to 4. These unregistered four objects were actually recommended to be instantiated, but failed to be instantiated because their positions were not successfully updated due to their too low recognition rates.

The experimental results reveal that the proposed method makes it possible to register object instances robustly even with an imperfect vision sensor. However, not every object was registered, regardless of object recognition rates. Unregistered objects, such as `Pot`, `GasBurner` and `WallClock` were likely difficult to register because of their poor recognition rates. Generally, the proposed method is not able to register objects with a recognition rate less than 52.9%($\gamma = 4$). Otherwise, the experimental results reveal that the method can determine misidentifications well. This means that we would better use a visual recognition system of an object recognition rate better than 52.9%. Then, the set of robot knowledge are reliable and have few false positives which make useless knowledge.

7 Conclusions

In this paper, we proposed a robust RoKI method for use under conditions of imperfect object recognition. The method uses temporal reasoning to check the validity of relationships between intervals and statistical reasoning to determine the CI of object recognition, moreover represents ontologically spatial relations between objects and semantic map. Determining failures from unreliable object recognition makes it possible to instantiate semantic knowledge dependably. In our novel approach, the robot verifies the recognized objects as true or not. The experimental results indicate that all false positives in recognition results were corrected. The proposed method had difficulty registering some objects with a recognition rate less than 52.9%($\gamma = 4$). In spite of this, the method can determine misidentifications well, and thus dependable semantic knowledge for service robots can be instantiated.

References

1. Thielscher M (2000) Representing the knowledge of a robot. In: The international conference on principles of knowledge representation and reasoning (KR-2000), Morgan Kaufmann, San Mateo, CA, 2000, pp 109–120
2. Hertzberg J (2008) Using semantic knowledge in robotics. Robot Auton Syst 56(11):875–877
3. Suh IH, Lim GH, Hwan W, Suh H (2007) Ontology-based multi-layered robot knowledge framework (OMRKF) for robot intelligence. In: IEEE/RSJ international conference on intelligent robots and systems 2007, San Diego, CA, USA, Oct 29–Nov 2, 2007, pp 429–436
4. Russell SJ, Norvig P (2003) Artificial intelligence: a modern approach, 2nd edn. Pearson Education. Inc
5. Lee DS, Lim GH, Suh IH (2009) Rule-based object detection for robust instance registration in robot-centered knowledge framework. In: International symposium on artificial life and robotics (AROB 12th '07), Beppu, Japan
6. Munich ME, Ostrowski J, Pirjanian P (2005) ERSP: a software platform and architecture for the service robotics industry. In: Intelligent robots and systems, 2005 (IROS 2005). 2005 IEEE/RSJ international conference, pp 460–467
7. Son TC, Tu PH, Baral C (2004) Planning with sensing actions and incomplete information using logic programming. In: Proceedings LPNMR-2004. Lecture notes in computer science/lecture notes in artificial intelligence, vol 2923. Springer, Berlin, Germany, pp 261–274
8. Obeid N, Rao RB (2007) A step toward a universal theory of failure handling. Diagnostyka 1(41):5–14
9. Thrun S, Burgar W, Fox D (2002) Probabilistic robotics. The MIT Press, ACM, New York, NY, USA
10. Shmulevich I, Dougherty ER, Kim S, Zhang W (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics, Oxford University Press, pp 261–274
11. Jahanian H, Hossein-Zadeh GA, Soltanian-Zadeh H, Ardekani BA (2004) Controlling the false positive rate in fuzzy clustering using randomization: application to fMRI activation detection. Magn Reson Imaging 22:631–638
12. Lim GH, Suh IH (2008) weighted Action-coupled Semantic Network (wASN) for robot intelligence. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS), Nice, France
13. Kyoshgoftaar TM (2004) Rule-based noise detection for software measurement data. In: IEEE international conference on information reuse and integration 2004. IRI 2004, Las Vegas, Nevada, USA, pp 302–307
14. Yang J, Liu J, Wang J, Sii H, Wang H (2006) A generic rule-base inference methodology using the evidential reasoning approach—RIMER. IEEE Trans Syst Man Cybern (Part A: Syst Hum) 36:266–285
15. Loyer Y, Straccia U (2003) Default knowledge in logic programs with uncertainty. In: International conference on logic programming, vol 2916 of LNCS, Mumbai, India, pp 466–480
16. Ding Z, Peng Y (2004) A probabilistic extension to ontology language OWL. In: Proceedings of the 37th Hawaii international conference on system sciences-2004, vol 4, p 40111a
17. Smith MK, Welty C, McGuinness DL (2004) OWL web ontology language guide. W3C recommendation, vol 10 (2004)
18. Brachman R, Levesque H (2004) Knowledge representation and reasoning. Morgan Kaufmann Publishers, San Francisco
19. Allen JF (1991) Planning as temporal reasoning. In: The second international conference on principles of knowledge representation and reasoning, Cambridge, MA, April, 1991
20. Heard A, Pensky M (2006) Confidence intervals for reliability and quantile functions with application to NASA space flight data. IEEE Trans Reliab 2006 55:591–601
21. Baader F, Calvanese D, Mcfuinness D, Nardi D, Patel-schneider P (2003) The description logic handbook, chap 2. Cambridge university press
22. Chuho Y, Suh IH, Lim GH, Jeong S, Choi B (2008) Cognitive representation and Bayesian model of spatial object contexts for robot localization. In: 15th international conference on neural information processing of the Asia-Pacific neural network assembly (ICONIP 2008), Auckland, New Zealand