

Research Article

Automated Space Classification for Network Robots in Ubiquitous Environments

Jiwon Choi,¹ Seoungjae Cho,² Phuong Chu,² Hoang Vu,² Kyhyun Um,² and Kyungeun Cho²

¹3D Systems Korea, Inc., Seoul 135-917, Republic of Korea

²Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 100-715, Republic of Korea

Correspondence should be addressed to Kyungeun Cho; cke@dongguk.edu

Received 20 September 2014; Accepted 27 November 2014

Academic Editor: Han-Chieh Chao

Copyright © 2015 Jiwon Choi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network robots provide services to users in smart spaces while being connected to ubiquitous instruments through wireless networks in ubiquitous environments. For more effective behavior planning of network robots, it is necessary to reduce the state space by recognizing a smart space as a set of spaces. This paper proposes a space classification algorithm based on automatic graph generation and naive Bayes classification. The proposed algorithm first filters spaces in order of priority using automatically generated graphs, thereby minimizing the number of tasks that need to be predefined by a human. The filtered spaces then induce the final space classification result using naive Bayes space classification. The results of experiments conducted using virtual agents in virtual environments indicate that the performance of the proposed algorithm is better than that of conventional naive Bayes space classification.

1. Introduction

The states of ubiquitous instruments, the daily supplies connected through the Internet of Things (IoT), are integrally managed in ubiquitous environments. In such environments, network robots can be operated in smart spaces via wireless networks. A network robot can effectively interact with ubiquitous instruments based on IoT and provide services required by senior citizens or patients in a smart space.

However, a space classification approach in which a smart space is reduced and identified as a space unit is required to execute behavior planning in real time for more practical services by network robots. Graph-based space classification and naive Bayes space classification are among a variety of space classification approaches that utilize objects identified in the environment [1, 2]. Graph-based space classification classifies a space using a graph predefined by a human being. In this approach, the graph expresses the inclusion relationship between the space and multiple objects; the robot searches the graph to determine the objects identified in the environment and the graph returns a space in which the

objects are included as the classification result. Naive Bayes space classification classifies a space using the appearance ratio of an object identified in various spaces. It requires a process that saves the appearance ratio in a knowledge base for classification. In this approach, the robot searches the knowledge base to ascertain the appearance ratio of objects identified in the environment and the knowledge base returns the space with the highest appearance ratio as the classification result.

In graph-based space classification, the graph is predefined by a human being. Thus, it includes the subjective thoughts of the human being predefined the graph and so the classification results vary. Naive Bayes classification needs to calculate and compare the probabilities of all spaces that can be classified. The problem with this calculation is that the number of operations is proportional to the number of spaces that can be classified. This paper proposes an approach that classifies the space efficiently by solving the problem above.

This paper makes three primary contributions. The first contribution involves classification of the space by subdividing it. Whereas conventional space classification approaches

divide a space into rooms and corridors [3], this paper proposes an approach in which a virtual agent is used to subdivide and classify the space. In virtual environments, virtual agents can be effectively simulated to verify an approach in this paper [4–6]. In the second contribution, the number of tasks predefined by a human being is reduced. Conventional space classification approaches classify the space by using a human being to predefine the relationship between the space and multiple objects via a graph. In such cases, the classification results vary according to the human being. Accordingly, it is difficult to secure consistent space classification performance. To rectify this situation, this paper minimizes the number of tasks predefined by a human being. Further, this paper examines how to improve the classification performance of the naive Bayes classifier-based space classification. Approaches that use the naive Bayes classifier use statistical data, resulting in them being more objective than approaches that use the relationship between a space and an object predefined by a human being. However, this type of classification requires more time for classification than graph-based space classification. Thus, it is necessary to speed up the execution of naive Bayes classifier-based space classification.

This paper proposes an approach that automatically generates a graph using the appearance ratio of objects without predefinition by a human being in the graph-based space classification. In the proposed approach, the space with the highest possibility is extracted by filtering the spaces that can be classified using the automatically generated graph. The spaces extracted by filtering are then classified using naive Bayes space classification and the final result identified. By utilizing our approach, the quantity of data inquiry is decreased to 44.75% in average, compared to existing naive Bayes space classification based approach. As our approach can perform space classification using much less operations than conventional approach, it is suitable to be performed in real time.

The remainder of this paper is organized as follows: Section 2 discusses previous research related to space classification and describes how they differ from the approach proposed in this paper. Section 3 describes the overall structure of the proposed approach. Section 4 explains the approach used to automatically generate a graph via probability. Section 5 discusses graph-based priority space classification. Section 6 compares by means of experiments naive Bayes space classification and the proposed approach and analyzes the results. Section 7 presents concluding remarks and outlines possible further study.

2. Related Work

This section discusses previous research conducted on space classification and primarily analyzes the advantages and disadvantages of various space classification methods applicable to virtual agents. Space classification research can be classified into five main approaches: graph-based space classification, space classification using the naive Bayes classifier, predicate logic-based space classification, robot action

pattern and image pattern-based space classification, and place recognition-based space classification.

Graph-based space classification classifies spaces using a graph. Its prerequisite is the classification of a room and a corridor [1]. A graph is used to classify the room and expresses the relationship between the predefined space and an object [7]. The result of space classification using a graph presents the interaction with a robot using Hybrid Logics Dependency Semantics (HLDS) [8]. In graph-based space classification, spaces are classified by ascertaining the space where an object belongs based on a graph search algorithm. For example, when a coffee machine is identified, it is included in a kitchen object group. The kitchen object group is included in the space “kitchen.” Accordingly, the inference result on the space where the virtual agent is located is calculated as the kitchen. Graph-based space classification requires less time for classification than other approaches because it classifies a space using a graph search without other operations. However, it also uses a number of things predefined by a human being, including kind of space, kind of object, and their correlation. The problems that may result from this predefinition are largely classified into three kinds. Graph-based space classification is faster than other approaches, but its drawback is that various parameters have to be predefined by a human being. Next, let us look at the kind of objects that are predefined. If there are obscure standards on the kinds of objects used when predefining them and the objects searched in an environment do not exist on the graph data, the spaces cannot be classified. The final problem is the objectivity of definition on the relationship between a space and an object. In graph-based space classification, a human being directly predefines the relationship between a space and an object. Such a graph may include the subjectivity of the producer and may therefore be less objective.

Naive Bayes classifier- (NBC-) based space classification utilizes probability [2]. A space is classified by the probability in which space an object can be identified the most, after learning the probability of the space where the object is identified in advance [9]. For example, when a bed is identified, suppose that the probability of a bed being identified in a bedroom is 0.7 and that in a living room is 0.1. The space where a virtual agent is located will return a bedroom as a result of inference by the classifier. NBC-based classification enables more accurate classification with more classification standards [10]. More accurate classification results can be acquired by collecting and classifying the probability depending on the number of objects and the probability depending on the distance, as well as the appearance ratio of objects [11]. NBC-based space classification does not have the problem that besets graph-based space classification in which objectivity of results after space classification is not secured because the graph is defined by a human being. However, NBC-based space classification also has three major problems. First, the space that can be classified needs to be calculated. Because NBC-based space classification calculates and compares all spaces that can be classified, increases in the number of spaces that can be classified result in increases in the classification time. The next problem involves how the

same kinds of objects are predefined. The last problem is that they require separate learning processes. Graph-based space classification can be applied only provided that there is a space and a predefined graph, NBC-based space classification requires supervised learning.

Predicate logic-based space classification predicatively predefines the objects in a space to classify the space [12]. A predicative expression describes the relationship between objects as well as the relationship between the space and objects. This approach enables an agent to establish a detailed plan [13]. The possibility of making a robot move from the present space to another space using predicate logic-based space classification has also been explored [14, 15]. Let us suppose that a robot has the goal “find milk.” In this case, a virtual agent will try to find a refrigerator, the object containing milk. To find the “refrigerator” object, the space “kitchen,” in which the refrigerator is located, will be searched. The agent plans and executes the actions—going to the kitchen, opening the refrigerator, and getting milk. This approach enables detailed planning by defining the relationship between objects. However, the problems associated with predicate logic-based space classification are virtually the same as those associated with graph-based space classification: a number of things are predefined by a human being, the kinds of objects have to be predefined, and objectivity is compromised in defining the relationship between a space and an object and between objects. In addition to the objectivity problem, predicate logic-based space classification may not be able to express all kinds of variables from the relationship between objects. Furthermore, when it can express such variables, the knowledge base needed is extensive and so a significant amount of time is required for space classification.

Space classification using robot action patterns uses the action patterns of a robot in a specific space for space classification [3]. The approach proposed in this paper classifies a room, a corridor, and a hall using six kinds of robot actions [3]: run in accordance with a predefined action pattern, turn right in place, turn left in place, go backward, stop, and crash sensor detection ability. The robot turns left and right in place to detect crashes and avoid obstacles in a room with the smallest space for movement among three spaces. The robot goes backward when it detects a crash to avoid a wall in a corridor when the corridor is relatively narrow. It executes only pattern actions in a hall because the hall is wide and does not go backward or detect a crash. Space classification using robot action patterns analyzes the patterns of a robot that vary on the features of a space for space classification. The advantage of this approach is that it does not require vision sensors for space classification, unlike other approaches. However, it cannot classify the space by segmenting a room.

Space classification using image patterns uses the directional histogram, configured using the images acquired by executing the rotational pattern of a robot, identified in a space [3]. A straight pattern with an angle in the range $0-2\pi$ is extracted from the images acquired to classify a space. When analyzing image patterns, a room gets the directional histogram from a number of angles because of the number

of objects in it. A corridor extracts a very simple directional histogram. A hall has a more directional histogram than a corridor but less than a room. Space classification using image patterns classifies a space without the object recognition step. However, it cannot classify a space by segmenting a room in the same way as space classification using a robot action pattern.

In place recognition-based space classification, a human being directly generates a topological map describing the environment using a map editor and designates the relationship between an image taken by Omnicam and the space. On the basis of the map and relationship established by the human being, the present space is classified by comparing the image taken moving around in the environment and the database [16]. Place recognition-based space classification can classify a space without the object recognition process and enables easy identification of the location of a virtual agent in the entire environment because the topological map is matched to the environment in advance. However, a human being is needed to create the topological map and match it to images. The most serious disadvantage is that, unlike other approaches, it cannot be applied to new environments. Place recognition-based space classification requires the implementation of an image database related to the environment and production of a topological map.

In recent space classification research, a variety of approaches depending on environment have been proposed: graph-based space classification using a graph defined by a human being, NBC-based space classification collecting and classifying the appearance ratio of objects, predicate logic-based space classification describing objects and their relationship, space classification using robot action pattern enabling space classification without vision sensor, space classification using image patterns without the object recognition process, and space recognition identifying the present location by creating a database with the images taken in the space and matching them with the images taken in real time. While a space can be classified using these conventional approaches, most of them depend substantially on information predefined by a human being.

Consequently, this paper focuses on NBC-based space classification, which has fewer tasks to predefine, and graph-based space classification, which has the lowest number of calculation requirements. The approach proposed in this paper is differentiated from previous approaches in two aspects.

First, the proposed approach reduces the number of tasks that need to be predefined by a human being, as compared to conventional approaches. In most conventional approaches, a significant number of tasks are predefined by a human being. The approach proposed in this paper uses graph-based space classification to automatically create a graph without any tasks being predefined.

Second, the proposed approach improves the performance of NBC-based space classification. NBC-based space classification requires a significant amount of time for space classification in order to calculate the probabilities of all spaces. The proposed approach conducts classification more

quickly than NBC-based space classification by calculating only the probability of specific spaces.

3. Overall Structure of Space Classification Using Automatically Generated Graph and Normalized Probability

The overall structure of the approach proposed in this paper comprises an automatic probability based graph generation module and a priority space classification module that uses the normalized probability and the generated graph.

The automatic probability based graph generation module generates the graph by collecting the object probability (appearance ratio) identified in a space. The first knowledge base of a virtual agent has no data so that it can execute the task to classify a space. Space classification using the naive Bayes classifier requires a process in which the number of appearances of an object in a space is identified and saved in the knowledge base. The approach proposed in this paper automatically creates a graph using the appearance ratio of objects acquired by searching a space based on probability. The graph describes the relationship between an object and a space and is connected to the space with the highest appearance ratio of an object. The graph automatically generated using the appearance ratio varies according to the environment in which the appearance ratio is acquired. The graph varies according to the environment to be investigated. Thus, the closer the environment is to the real world, the closer the relationship between an object and a space acquired on the graph is to the real world.

In automatic probability based graph generation, a process that normalizes the probability and creates a graph is executed. The process reduces the quantity of data required for classification. When the probability of an object in a space is acquired, the sum of the probability of the object identified in each space is irregular. The sum of the probability is normalized as one and the ratio to total probability can be estimated only with the probability of one space. Since only one standard value is requested and compared without requesting and comparing the probability of all spaces, the quantity of data required for classification is reduced.

Priority space classification uses the graph created by the automatic probability based graph generation module and the standard value normalizing the probability. The space, which is highly possible to convert as the result of space classification, is filtered before space classification by the automatically generated graph. In accordance with the filtering result, the space classification is executed from the space with a higher possibility. A space is classified using the standard value according to the priority of the space acquired by filtering. The standard value of the space with higher priority of an identified object is requested and compared with the total standard value. If the value is greater than the total standard value, the relevant space is returned as the result. However, if it is less than the total standard value, the standard value of the next priority object is requested and the process above is continued.

4. Automatic Probability Based Graph Generation

Automatic probability based graph generation generates a graph by collecting the probability of an object being identified in a space (appearance ratio). To create the graph, the probability of the object being identified in each space (appearance ratio) is collected in various environments using the data collection module. The collected probability is then saved in the knowledge base and the generation module normalizes the probability of the saved knowledge base and generates a graph. The result from the automatic probability based graph generation module is saved as the automatically generated graph and the standard value normalizing probability in the knowledge base.

Section 4.1 describes the structure of the automatic probability based graph generation system. Section 4.2 explains the data collection module that saves the probability of the objects identified in diverse spaces. Section 4.3 describes the graph generation module that normalizes the probability of the objects identified and creates a graph.

4.1. Structure of the Automatic Probability Based Graph Generation System. The automatic probability based graph generation system comprises a data collection module, a generation module, and a knowledge base, as shown in Figure 1.

The data collection module uses the structure of the space classified using naive Bayes classifier as is. This module recognizes the objects existing in the environment and updates the probability of objects identified in the space to the knowledge base.

The generation module normalizes the object probability data of the knowledge base collected using the data collection module and generates a graph.

4.2. Data Collection Module. The data collection module comprises three modules that save the object identification probability.

The first module is the object identification module. This module obtains the image data in the environment recognized by the virtual agent and recognizes the objects using an image processing technique that incorporates the scale invariant feature transform (SIFT) algorithm to check what kinds of objects exist in the image data. The focus of this paper is on the space classification algorithm; hence, the image processing technique is beyond its scope. In this paper, an object is identified when it is included in the sight of a virtual agent. If an object is included in the sight of a virtual agent, the object type is recognized and the results are delivered to the space classification module.

The second module is the NBC-based space classification module. This module classifies the space containing the virtual agent by calculating the probability of an object using the object recognition module. It receives the list of identified objects from the object recognition module, classifies the space, and delivers the classification results. The data collection module includes the naive Bayes classifier to identify any change in classification accuracy in accordance

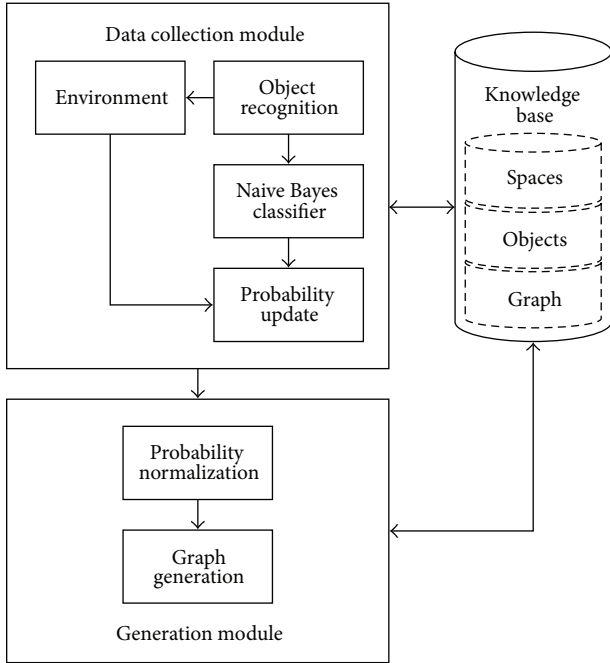


FIGURE 1: Structure of the automatic probability based graph generation system.

with the update probability of an object in the knowledge base using the probability update module:

$$R = \arg \max_k \left(\prod_{i=1}^n P(o_i | s_k) \right). \quad (1)$$

Equation (1) defines NBC-based space classification. With the list of identified objects o_i , k , the space with the highest value after multiplication of probability that an object is in the space s_k , is saved in the classification result R . n is the number of objects.

The third module is the probability update module. This module receives the results from the space classification module, ascertains whether the results are matched to a real space, and updates the probability of identified objects. A virtual agent at first has no data in the knowledge base so that the space classification results are not proper. The classifier can properly complete the classification by saving the appearance ratio of objects in a space using the update module and repeating the processes above in a variety of environments.

4.3. Generation Module. The generation module operates after the data collection module. It comprises two modules: the probability normalization module and the graph generation module.

The probability normalization module normalizes the sum of the probabilities of objects in each space to 1.0. As shown in (2), the probability with which an object is identified in each space varies before probability normalization and the sum is also inconsistent. Even though the probability of identification in each space is higher, the ratio of each space

TABLE 1: Probability before normalization.

	Bedroom	Living room	Kitchen	Bathroom	Total
Bed	0.9	0.1	0.0	0.0	1.0
Dressing table	0.7	0.0	0.0	0.0	0.7
Sofa	0.2	0.8	0.0	0.0	1.0
Table	0.3	0.3	0.5	0.0	1.1
Desk	0.3	0.3	0.0	0.0	0.6
Bathtub	0.0	0.0	0.0	1.0	1.0

TABLE 2: Probability after normalization.

	Bedroom	Living room	Kitchen	Bathroom	Total
Bed	0.9	0.1	0.0	0.0	1.0
Dressing table	1.0	0.0	0.0	0.0	1.0
Sofa	0.2	0.8	0.0	0.0	1.0
Table	0.27	0.27	0.46	0.0	1.0
Desk	0.5	0.5	0.0	0.0	1.0
Bathtub	0.0	0.0	0.0	1.0	1.0

to the entire space varies on the sum. This paper normalizes the probability of each space by the ratio to 1.0:

$$NP_{ky} = \frac{P(o_y | s_k)}{\sum_{i=1}^n P(o_y | s_i)}. \quad (2)$$

Equation (2) normalizes the probability of an object. NP_{ky} , the normalization result, is calculated by dividing $P(o_y | s_k)$, the probability of identification of o_y in the space s_k , by $\sum_{i=1}^n P(o_y | s_i)$, the sum of $P(o_y | s_i)$, which is the probability of identification of o_y in the space s_i . n is the number of spaces.

As shown in Table 1, the sum of the probabilities with which an object is identified in each space is irregular.

Probability normalization aims at space classification only by identifying the probability in one space without the probability in other spaces, as normalizing the sum of irregular probability identifying the objects is 1.0. For example, when the dressing table in Table 2 is identified, the existing algorithm searches and compares the probabilities of all spaces. When the probability of bedroom is searched using the probability normalization process, the probability of the area being the bedroom and the opposite probability obtained by deducting the probability of bedroom are compared. Consequently, the space can be classified without identifying and comparing the probabilities of all spaces.

The graph generation module generates a graph using the standard value. Unlike conventional graph-based space classification approaches, which require a human being to predefine the graph, the approach in this paper automatically generates the graph using the normalized probability value with which an object is identified. The graph is generated by

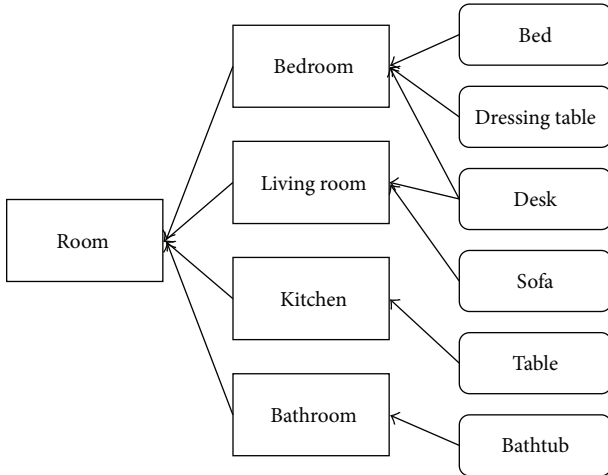


FIGURE 2: The graph generation process.

connecting the space with the highest value (Max) among the standard values of objects to an object:

$$G_y = \arg \max_k (NP_{ky}). \quad (3)$$

The graph is generated using (3). \max_k , the space with the highest appearance ratio for an object y using the standard value NP_{ky} , is saved as the result in G_y and a graph is generated.

Figure 2 illustrates how the graph is generated. A space and an object are connected in accordance with the highest value among the standard values. If the highest values are found in the same space, one object, such as a desk, is connected to two spaces.

5. Graph-Based Priority Space Classification

Graph-based priority space classification requires advance automatic probability based graph generation. A graph is automatically generated using probability that needs to be normalized. Graph-based priority space classification classifies the space using a graph and a standard probability. It classifies the space on the assumption that there is no feedback from the environment. It also executes filtering using a graph and determines the priority of the space with higher possibility to be returned as the classification result. Further, following graph filtering, it classifies the space using the naive Bayes classifier in accordance with the order of priority.

This section proposes graph-based priority space classification, which reduces the time for classification compared to existing NBC-based approaches that use generated graphs. Section 5.1 describes the overall structure of the graph-based priority space classification system. Section 5.2 explains the graph filtering process. Section 5.3 presents the proposed approach based on priority naive Bayes classifier using the results of graph filtering.

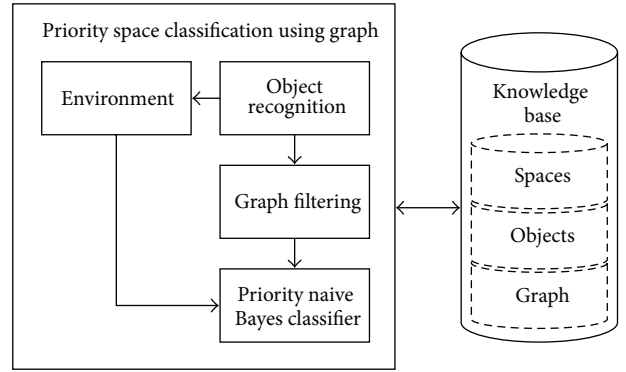


FIGURE 3: Structure of the graph-based priority space classification system.

```

For j = 0 to n
  Fj = arg maxk (countk(Goi))
  Except (arg maxk)
End For

```

ALGORITHM 1: Decision-making code in graph filtering.

5.1. Structure of the Graph-Based Priority Space Classification System. The graph-based priority space classification system comprises an object identification module, a graph filtering module, and the priority naive Bayes classification module, as shown in Figure 3.

The object identification module identifies the object in the sight of the virtual agent. The graph filtering module receives data describing the identified object from the object identification module, searches for the graph, and filters it. In accordance with the filtering results, the spaces with higher possibility are arranged in order and the results returned. The priority naive Bayes classification module receives the graph filtering result, calculates the probability from the space with higher possibility to be returned in accordance with the classification results, and delivers the space classification results.

5.2. Graph Filtering. The graph filtering module receives the data associated with an identified object from the object identification module and filters the space in the order of higher priority to be returned as the final classification result. The filtering result is delivered to the priority naive Bayes classification module. Graph filtering searches for the identified object in a graph and determines the order of calculation depending on which space is connected. Algorithm 1 illustrates the graph filtering process.

o_i , the list of objects identified in the object identification module, is searched in graph G_{o_i} . In accordance with the graph search, an object o_i and the space k connected to the object are counted as $\text{count}_k(G_{o_i})$. Subsequently, argmax_k , the space with the highest count, is returned as the result. The process above continues according to the number of spaces. Then, the spaces are saved in the filtering result F_j , in order

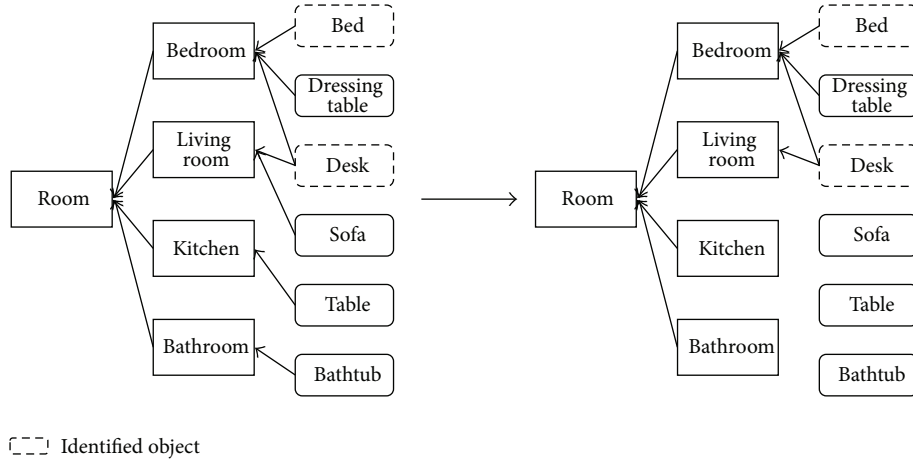


FIGURE 4: Example of graph filtering.

from the space with the highest count to the space with the lowest count. Figure 4 depicts an example of graph filtering.

If a bed and a desk object are identified, both objects are searched in the graph to find the space connected with each object. In accordance with the search results, the bed is connected to a bedroom and the desk to a living room. After completing the search, the connected spaces are counted and filtered. In accordance with the filtering, the space is filtered in the order of bedroom, living room, kitchen, and bathroom.

5.3. Priority Naive Bayes Classification. Priority naive Bayes classification classifies a space using the graph filtering result and the standard value. The standard value compares the probability of a specific space and total standard value without the specific space ($1 - \text{standard value}$). When the space with the highest probability is returned in accordance with the graph filtering and space classification, only the standard value of one space is searched and the space can be classified. Algorithm 2 shows the decision-making code in the priority naive Bayes classification process. Priority naive Bayes classification compares the standard value of space s_{f_j} among graph filtering result F_j and the standard value of s'_{f_j} , which deducts s_{f_j} from the total probability. If space s_{f_j} is larger, it is returned as the classification result and space classification ends:

$$N(s_k) = \prod_{i=1}^n N_{ko_i}. \quad (4)$$

Equation (4) is used to calculate the standard value of a specific space. The calculation result for space $N(s_k)$ is acquired by multiplying the standard value of space k when an object o_i is identified:

$$N(s'_k) = \prod_{i=1}^n (1 - N_{ko_i}). \quad (5)$$

Equation (5) is used to calculate the total probability without the standard probability of a specific space. Since total

```

For j = 0 to n
  If  $N(s_{f_j}) > N(s'_{f_j})$ 
     $R = s_{f_j}$ 
    Return R
  End If
End For

```

ALGORITHM 2: Decision-making code for priority naive Bayes classification.

probability is normalized to one, the standard probability of a specific space is deducted from the total probability. When an object o_i is identified, the standard value of space k is deducted from the total probability of one. Next, the result of calculation of space $N(s'_k)$ is acquired via multiplication.

Figure 5 is an example of priority naive Bayes classification. The probability is calculated with the space having the greater possibility returned as the classification result using the graph filtering result.

6. Experiment and Analysis

This section compares the experimental results obtained using the graph-based priority space classification proposed in this paper with those obtained using the conventional naive Bayes space classification. Experiments were executed in a 3D virtual environment generated based on the interior design book. The experiments analyzed the performance of the existing classification approach and proposed a classification approach on the basis of two standards: quantity of data inquiry from knowledge base for implementation of classification algorithm and accuracy on whether the classification result was identical to the interior design book.

Section 6.1 describes the experimental objective and approaches. Section 6.2 introduces the experimental system and summarizes the experiments. Section 6.3 analyzes the experimental results.

Bedroom (2)	Living room (1)	Kitchen (0)	Bathroom (0)
Bedroom (2)	Compare	Not bedroom	
Living room (1)	Compare	Not living room	
Kitchen (0)	Compare	Not kitchen	
Bathroom (0)	Compare	Not bathroom	

FIGURE 5: Space classification by priority naive Bayes classifier.

6.1. Experiment Objective and Approaches. The aim of the experiments was to verify that graph-based priority space classification has less data inquiry than the conventional naive Bayes space classification with the same classification accuracy.

Two hundred virtual environments were produced in an invisible file to identify the probability with which an object is identified in a space. In the experiments, 10 kinds of 3D virtual environments were created, listed in Table 3, using 3DsMax. Each environment had four spaces, a bed, a living room, a kitchen, and a bathroom.

The experiments assumed 10, 50, 100, 150, and 200 virtual spaces to collect the probability with which an object is identified in a space and generate a graph. Ten kinds of 3D virtual environments were used to identify changes in the accuracy and the quantity of data inquiry carried out by the conventional naive Bayes classification and the graph-based priority space classification. Subsequently, the probability of object identification was collected from 200 spaces and a graph generated. Finally, the conventional naive Bayes classification and the graph-based priority space classification were compared in each 3D virtual environment.

6.2. Experimental System and Description. The experiments were implemented in the 3D virtual environment using the Gamebryo engine. The Gamebryo-based 3D virtual environment was configured as shown in Figure 6.

In the experimental environment, a virtual agent moved to each space with the objects in its sight displayed on the bottom left. A door was used to determine whether the virtual agent had moved to another space. When the virtual agent went to another space through a door, the data on the object identified in the previous space was initialized. The virtual agent identified the objects when they were included in the area of frustum indicating the sight of the character. (Frustum is inserted to describe the sight and was not seen during the experiment.) When objects in a space were completely identified, the space was either classified using the existing naive Bayes space classification or graph-based priority space

TABLE 3: The experimental 3D virtual environments.

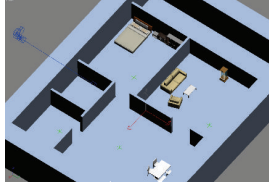
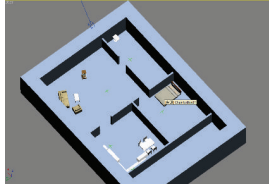
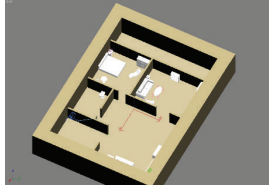
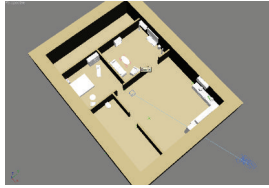
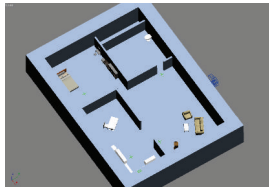


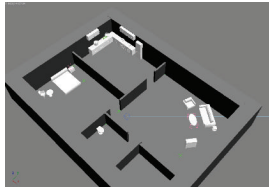
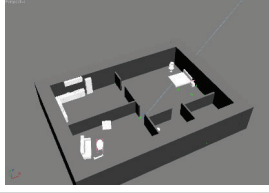

Virtual environment number	Configuration image
1	
2	
3	
4	
5	
6	
7	

TABLE 3: Continued.

Virtual environment number	Configuration image
8	
9	
10	

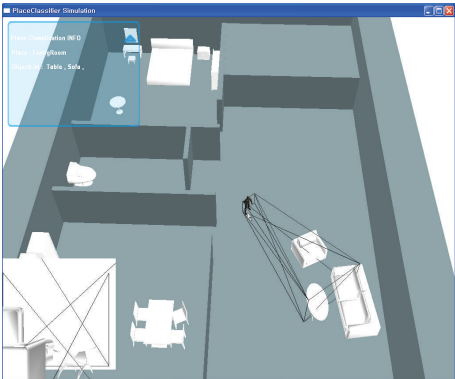


FIGURE 6: Gamebryo-based 3D virtual environment.

classification. The classification results are displayed in the top left of the screen.

6.3. *Experimental Analysis and Results.* The experimental results for the existing naive Bayes space classification and the graph-based priority space classification are divided into two kinds depending on the experimental approach. The results obtained for the experiment in which 10, 50, 100, 150, and 200 virtual spaces were made to generate a space and the performance in 10 kinds of 3D virtual environments compared are presented in Figures 7 and 8. The quantity of data inquiry and the accuracy were changed depending on the number of virtual spaces to generate a graph.

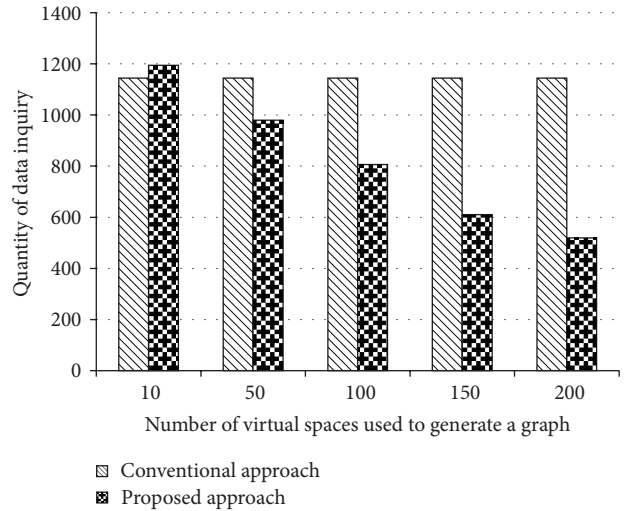


FIGURE 7: Change in the quantity of data inquiry depending on the number of virtual spaces.

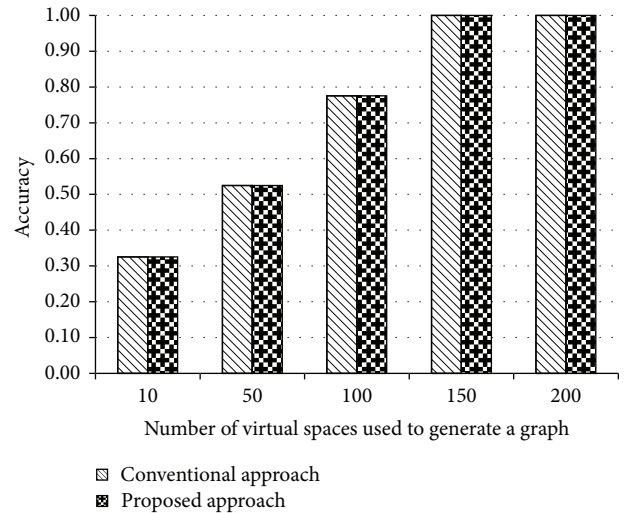


FIGURE 8: Change in accuracy in the virtual space.

Figure 7 illustrates the change in the quantity of data inquiry depending on the number of virtual spaces. The quantity of data inquiry increased by 104%, 85%, 70%, 53%, and 45% when the number of virtual spaces was 10, 50, 100, 150, and 200, respectively, compared to conventional naive Bayes space classification. As the number of virtual spaces in the probability based graph generation increased, the quantity of data inquiry in the graph-based priority space classification decreased compared to the conventional naive Bayes space classification.

Figure 8 presents the accuracy of space classification according to the number of virtual spaces. When the number of virtual spaces was 10, 50, 100, 150, and 200, the accuracy of space classification improved by 33%, 53%, 78%, 100%, and 100%, respectively. The conventional naive Bayes classification and the graph-based priority space classification show the same accuracy. As with the quantity of data inquiry,

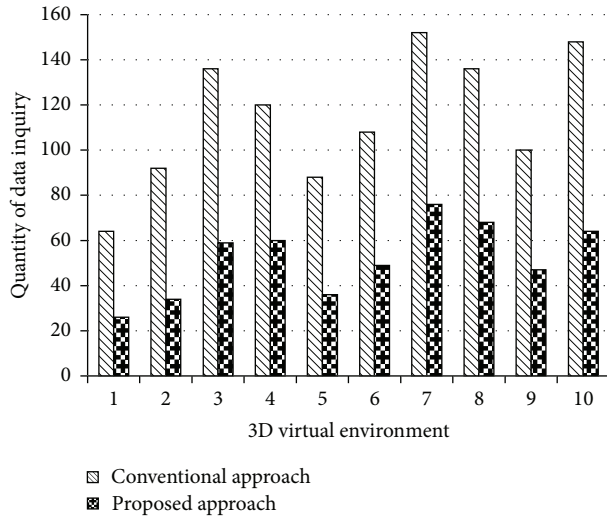


FIGURE 9: Quantity of data inquiry.

saving the probability of object identification and generating a graph in more virtual spaces improved the space classification accuracy.

Next, the performances in 200 virtual spaces and 10 kinds of 3D virtual environments were compared. The graph-based priority space classification proposed in this paper reduced the quantity of data inquiry with the same accuracy as the conventional naive Bayes space classification. In accordance with the experimental results, space classification was executed a total of 40 times because there were 10 kinds of 3D virtual environments, with four spaces for each environment.

Figure 9 shows the results from the implementation of conventional naive Bayes space classification (existing approach) and the graph-based priority space classification (proposed approach) in 3D virtual environments. In accordance with the experimental result, the space classification for the proposed approach was implemented with less data inquiry in all 10 kinds of environments than the conventional approach. The total quantity of data inquiry was 1,144 times in the conventional approach and reduced to 519 in the proposed approach.

Figure 10 presents the reduction ratio of the quantity of data inquiry in the proposed approach as compared to the conventional approach in the 3D virtual environment. The proposed approach reduced the quantity of data inquiry by 44.75% on average compared to the conventional approach.

Figure 11 presents the accuracy of the conventional and proposed approaches in the 3D virtual environments. Both approaches accurately classified the 10 virtual environments.

7. Conclusion

This paper proposed a space classification approach that uses standard probability and automatically generated graphs for virtual agents. The proposed approach is designed to satisfy real-time virtual environments; consequently, the proposed space classification generates a graph without predefinition.

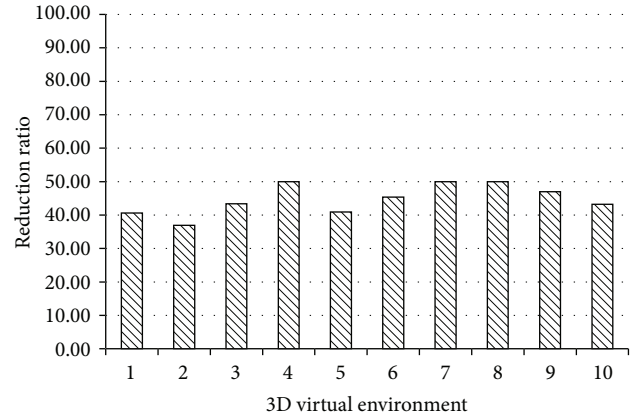


FIGURE 10: Reduction ratio in quantity of data inquiry.

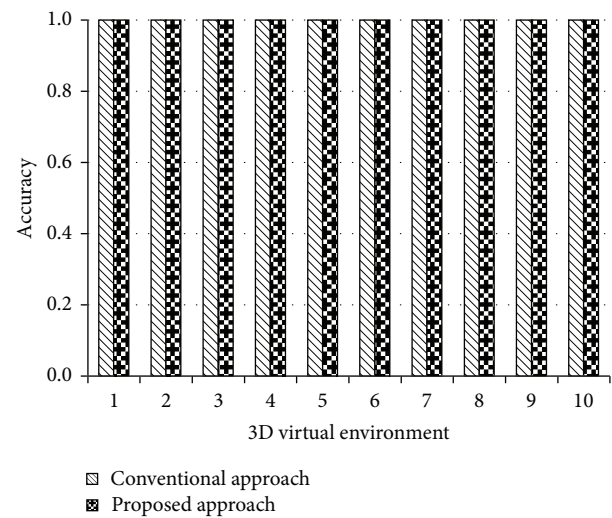


FIGURE 11: Accuracy of space classification.

The performance of the graph-based priority space classification was found to be better than that of conventional naive Bayes space classification.

The results of the experiments conducted indicate that the proposed approach reduces the quantity of data inquiry, the graph filtering results improved as the number of virtual spaces increased, compared to conventional naive Bayes space classification. The proposed approach reduced the quantity of data inquiry by 44.75% on average compared to the conventional approach in 10 kinds of 3D virtual environments. Further, even though the quantity of data inquiry was reduced, the classification accuracy remained the same as that of conventional naive Bayes space classification. The approach proposed in this paper can be applied to behavior planning for network robots in ubiquitous environments.

Further study that considers the relationship among objects for establishing more intelligent action plans on the basis of the space classification approach is needed. Furthermore, it is necessary to study how to apply the proposed algorithm in real environments. To expand the approach proposed in this work for facilitating it in more large and

complex smart spaces, a spatial feature should be considered additionally, as well as an object feature. Therefore, research of space classification algorithm based on machine learning (e.g., support vector machine algorithm) should be preceded, in order to apply learning and recognition of multidimensional feature vectors.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1021) supervised by the NIPA (National IT Industry Promotion Agency).

References

- [1] Ó. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.
- [2] J. H. Kang, *Probability-Based Place Classification in Indoor Environments Using Semantic Information and Information Gain*, Department of Electronics Computer Engineering, The Graduate School, Hanyang University, 2008.
- [3] Y. G. Oh, *Place Classification Using Robot Behavior and Vision in Indoor Environment*, Department of Electronics Computer Engineering, The Graduate School, Hanyang University, 2009.
- [4] M. Lim and Y. Lee, "A simulation model of object movement for evaluating the communication load in networked virtual environments," *Journal of Information Processing Systems*, vol. 9, no. 3, pp. 489–498, 2013.
- [5] M. Mozgovoy and R. Efimov, "WordBricks: a virtual language lab inspired by Scratch environment and dependency grammars," *Human-Centric Computing and Information Sciences*, vol. 3, article 5, 2013.
- [6] O. Mirzaei and M. R. T. Akbarzadeh, "A novel learning algorithm based on a multi-agent structure for solving multi-mode resource-constrained project scheduling problem," *Journal of Convergence*, vol. 4, no. 1, pp. 47–52, 2013.
- [7] H. Zender, O. Martínez Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.
- [8] Ó. M. Mozos, P. Jensfelt, H. Zender, G.-J. M. Kruijff, and W. Burgard, "From labels to semantics: an integrated system for conceptual spatial representations of indoor environments for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 33–40, Rome, Italy, April 2007.
- [9] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots—an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.
- [10] S. Vasudevan and R. Siegwart, "A Bayesian conceptualization of space for mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 715–720, November 2007.
- [11] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 522–537, 2008.
- [12] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 955–966, 2008.
- [13] A. Bouguerra, L. Karlsson, and A. Saffiotti, "Monitoring the execution of robot plans using semantic knowledge," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 942–954, 2008.
- [14] A. Bouguerra, L. Karlsson, and A. Saffiotti, "Handling uncertainty in semantic-knowledge based execution monitoring," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 437–443, San Diego, Calif, USA, November 2007.
- [15] A. Bouguerra, L. Karlsson, and A. Saffiotti, "Semantic knowledge-based execution monitoring for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3693–3698, April 2007.
- [16] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A realistic benchmark for visual indoor place recognition," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 81–96, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

