Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8567271, 18 pages
https://doi.org/10.1155/2020/8567271

Research Article

Visual Experience-Based Question Answering with Complex Multimodal Environments

Incheol Kim

Department of Computer Science, Kyonggi University, Suwon-si 16227, Republic of Korea

Correspondence should be addressed to Incheol Kim; kic@kyonggi.ac.kr

Received 13 August 2020; Revised 13 October 2020; Accepted 28 October 2020; Published 19 November 2020

Academic Editor: Jiayi Ma

This paper proposes a novel visual experience-based question answering problem (VEQA) and the corresponding dataset for embodied intelligence research that requires an agent to do actions, understand 3D scenes from successive partial input images, and answer natural language questions about its visual experiences in real time. Unlike the conventional visual question answering (VQA), the VEQA problem assumes both partial observability and dynamics of a complex multimodal environment. To address this VEQA problem, we propose a hybrid visual question answering system, VQAS, integrating a deep neural network-based scene graph generation model and a rule-based knowledge reasoning system. The proposed system can generate more accurate scene graphs for dynamic environments with some uncertainty. Moreover, it can answer complex questions through knowledge reasoning with rich background knowledge. Results of experiments using a photo-realistic 3D simulated environment, AI2-THOR, and the VEQA benchmark dataset prove the high performance of the proposed system.

1. Introduction

With the rapid developments in the deep learning technology, high-level image understanding problems are gaining increasing attention in computer-vision communities [1, 2]. Visual question answering (VQA) [3] is one of the most actively researched image understanding problems that requires the generation of correct answers to natural language questions about input images, as illustrated in Figure 1(a). This is a complex intelligence problem that requires both image and natural language understanding abilities.

However, existing VQA problems have a few limitations. First, it is difficult to understand the three-dimensional (3D) configuration of the entire environment because only one input image is provided with no distinction between indoor and outdoor surroundings. Consequently, the scope of questions covers only a part of the environment in space or time. Furthermore, many questions do not require commonsense or background knowledge and can be sufficiently answered by the understanding of the input image and question. These VQA problems do not consider the agent's body in the environment or interactions between the agent and environment. Therefore, unlike the real world, it is impossible to ask questions about the state of an agent when acquiring the input image and about the environmental change after the agent performs a specific action.

To overcome these limitations of the existing VQA problems, the present study proposes a visual experience-based question answering (VEQA) problem and a question answering system (VQAS) to solve the problem. The VEQA can be considered a type of an embodied VQA (EVQA) problem that has recently begun to be researched in the computer-vision field [4, 5]. The conventional EVQA problems assume that environmental changes do not exist except for changes in the agent position; however, in this study, the authors assumed that environmental changes are possible not only by the positional movement actions of the agent but also by manipulative actions such as picking up bread and opening the refrigerator. Furthermore, in the existing EVQA problems, the agent must plan the movement actions that it will perform to obtain the answer. However, the new VEQA problem is different from the EVQA problem in that the agent must perform a series of

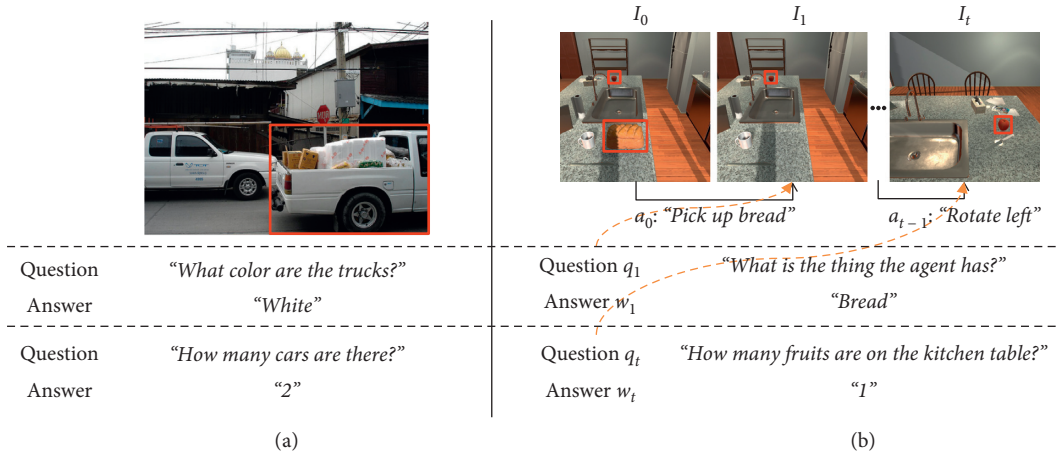

Figure 1: Visual question answering (VQA) and visual experience-based question answering (VEQA). (a) Example of VQA. (b) Example of VEQA.

actions that have been planned in advance and must answer questions about environmental state changes that it has experienced. Moreover, in the existing EVQA problems, no model about the agent's action is assumed, whereas in the VEQA problems the agent is assumed to have a probabilistic model about its actions in advance.

Figure 1(b) illustrates the proposed VEQA. As shown, while performing a series of actions $\langle a_0, \ldots, a_{t-1} \rangle$, the agent observes input images $\langle I_1, \ldots, I_t \rangle$ of the environment in the visible range. Furthermore, the range of questions, q_t, given at time t is limited to 3D configurations and the environmental state experienced by the agent through the input images $\langle I_0, \ldots, I_t \rangle$ until that time. In this example, immediately after performing action a_0 = "Pick up Bread," correct answer w_1 to question q_1 = "What is the thing the agent has" is "Bread." Furthermore, the VEQA problem requires separate commonsense knowledge in addition to proper understanding of the changing environmental state. For example, in Figure 1(b), to answer question q_t = How many fruits are on the kitchen table?," a separate commonsensical knowledge that the observed object "Apple" is a type of "Fruit" is required. The environmental states for question answering are represented in the 3D scene graph, as shown in Figure 2. The 3D scene graph is a knowledge graph consisting of objects in a 3D environment, object attributes, and spatial relationships between objects. To solve the VEQA, generation of the 3D scene graph is needed. Therefore, the answer to the question is made based on the 3D scene graph. Consequently, the VEQA is a problem of generating 3D scene graphs from the agent's visual experiences and answering the questions based on these 3D scene graphs.

Unlike the conventional VQA problem, the VEQA problem addressed in this study is designed for embodied intelligence in a photo-realistic 3D simulated environment close to the real world, which requires an agent to do navigations and actions, understand the entire 3D scenes from successive partial input images, and answer questions about the dynamic scenes. To deal with the VEQA problem, we suggest a structured representation to express the visual scene understanding of the dynamic 3D environment with a series of 3D scene graphs. A 3D scene graph captures objects and their spatial relationships in a given scene of the environment. The structured information represented in scene graph is useful for downstream tasks such as visual question answering. However, most existing scene graph generation models [6–9] generate a 2D scene graph from a single image. Therefore, they cannot model the entire 3D scene of the environment from a sequence of partial images, nor represent dynamic changes of scenes caused by the agent's navigation and action. To overcome such limitations of the conventional models, we propose a novel 3D scene graph generation model which can generate a series of 3D scene graphs from a sequence of partial input images in a dynamic environment. This model consists of the state recognition module and the state prediction module. While the former recognizes environmental states from input images based on the trained deep neural network, the latter predicts environmental changes caused by the agent's actions using the rule-based action models.

To address the VEQA problem, we also propose a knowledge reasoning system to answer questions about dynamic scenes of the environment based on 3D scene graphs. Some VEQA questions require deep background knowledge such as object hierarchies, which is beyond the shallow knowledge contained in 3D scene graphs generated on the fly from input images. However, the conventional visual question answering models based on pure deep neural network [10, 11] are not easy to utilize the structural information of 3D scene graphs. Moreover, the models are also hard to make use of background/prior knowledge of the environment. Different from the pure deep neural network-based models, the proposed knowledge reasoning system can use a rich knowledge source to answer questions by combining the shallow knowledge in 3D scene graphs with a large amount of prebuilt deep background knowledge.

The contributions of this paper are summarized as follows:

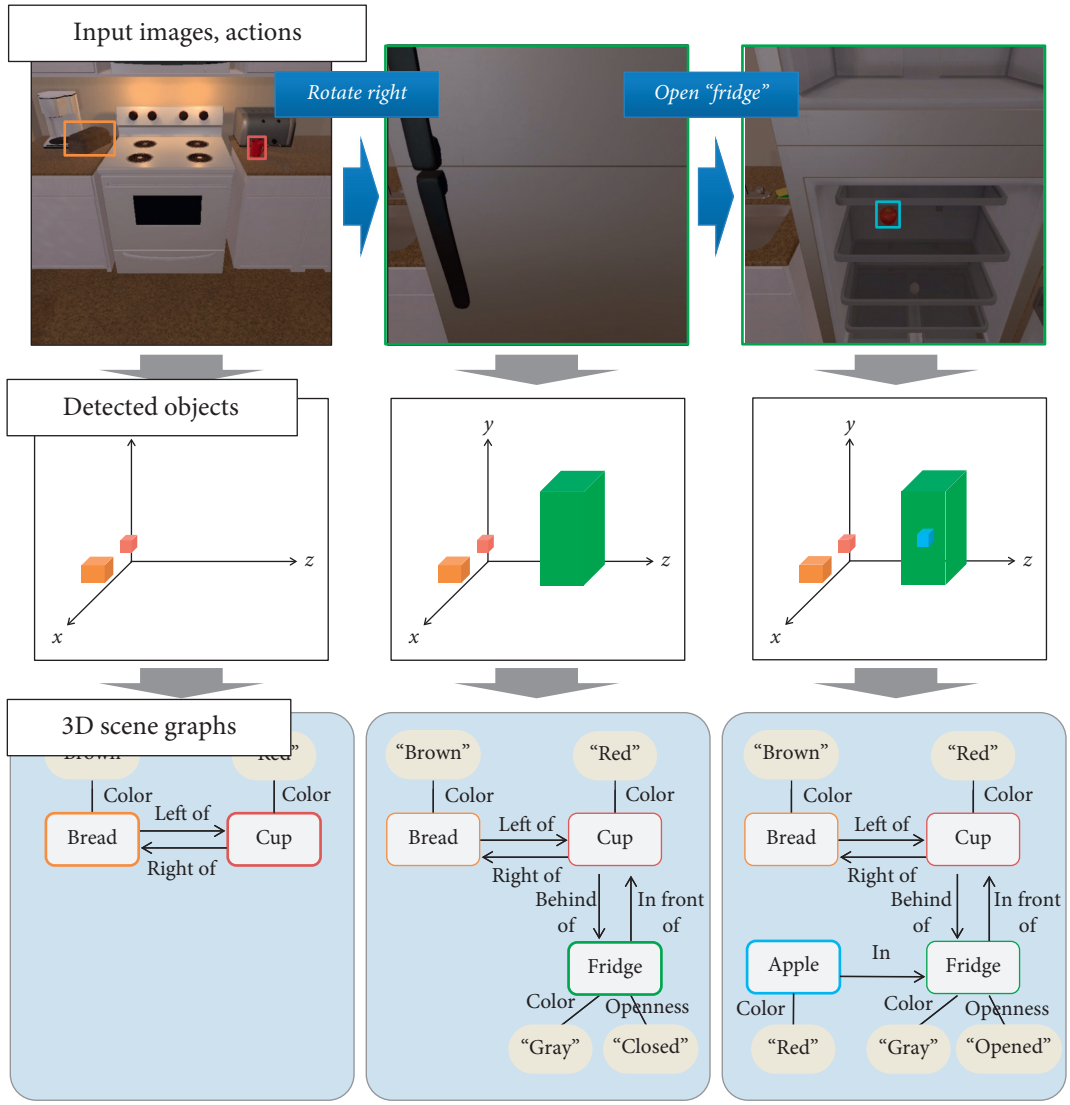

Figure 2: Example of 3D scene graph generation.

(1) We propose a novel VEQA problem and the corresponding dataset for embodied intelligence research that requires an agent to do actions, understand entire 3D scenes from successive partial input images, and answer natural language questions about the dynamic scenes in a photo-realistic 3D simulated environment.

(2) To address the VEQA problem, we propose a hybrid visual question answering system, VQAS, integrating a deep neural network-based scene graph generation model and a rule-based knowledge reasoning system.

(3) We propose a novel 3D scene graph generation (SGG) model which can generate a series of 3D scene graphs from successive partial input images in a dynamic environment. The proposed model can overcome the limitation of the conventional scene graph generation models building just a 2D scene graph from a single still image. The model also meets well the partial observability and dynamics of the VEQA environment.

(4) We also propose a knowledge reasoning system to answer natural language questions based on 3D scene graphs. Different from the pure deep neural network-based models, the proposed knowledge reasoning system can use a rich knowledge source to answer questions by combining the shallow knowledge in 3D scene graphs with a large amount of prebuilt deep background knowledge.

(5) The high performance of the proposed VQAS system is verified through a series of experiments using a photo-realistic 3D simulated environment, AI2-THOR, and the VEQA benchmark dataset.

2. Related Work

2.1. Scene Graph Generation. Scene graph generation involves the expression of scenes in images as knowledge in graph form [12, 13]. A scene graph generally consists of nodes representing the objects in the image and edges representing the relationships between objects [14]. This

scene graph can be effectively used in problems that require high-level image understanding such as image captioning and generation [15, 16].

Existing relevant works have generated 2D scene graphs to express objects in a single image and the relationships between the objects by matching them to a 2D space [6–9]. These works sequentially performed image-based object detection and relationship recognition between the objects and then expressed the results in one scene graph. In the object detection process, the faster R-CNN [17] was mainly used to recognize the position and class of objects in images based on the convolutional neural network. In the relationship recognition process, various features about the object region were used to determine the relationships between the detected objects. For accurate relationship recognition, researchers used various features, such as visual and spatial features, of each object. However, they did not examine 3D scene graphs that can express the 3D position of objects in images and their 3D spatial relationships.

Contrary to existing scene graphs studies, visual graphs from motion (VGfM) [18] generated a 3D scene graph consisting of 3D objects from multiple images and their spatial relationships. This study performed 2D object detection from input images and calculated the 3D object region by using the positions of objects in the images and the observer's pose. Further, the object classes and spatial relationships between the objects were derived from the features of the objects appearing in multiple images through the recurrent neural network. However, this method has the following constraints: all the objects to be recognized in each image must be captured and each image must represent the same environmental state. Therefore, VGfM [18] cannot be applied to partially observable and dynamically changing environmental conditions such as those in VEQA. To overcome these limitations, the present study proposes a method of correctly expressing the dynamic environmental state by expanding and upgrading the 3D scene graph according to the input of partial images.

2.2. Visual Question Answering. Existing studies on VQA proposed various deep neural network models to solve problems [4, 5]. Many existing studies extracted visual features by applying the convolutional neural network and extracted linguistic features by applying the recurrent neural network for natural language questions. Then, they combined these two features by applying an appropriate attention mechanism and trained it in an end-to-end method to generate correct answers to questions [19–24]. Some studies attempted to generate higher-quality answers by extracting high-level semantic features from input images [25–27]. Anderson et al. [25] conducted object detection in advance for input images to determine which object region to focus on in the answer-generation stage. Teney et al. [26] used graph-structured representation for VQA to detect objects in input images and determined similarity between the detected objects and the words constituting natural language questions. Through this process, they tried to generate answers by using the object features with high similarity with the question.

However, these VQA models only train the scenes in the input images as implicit knowledge embedded in the deep neural network. Therefore, the implicit knowledge obtained from images in this manner is difficult for humans to understand compared to explicit knowledge in symbolic logic or graph form. Furthermore, explaining the validity of the generated answer is difficult. Moreover, combining such knowledge with various external knowledge bases and using them for more in-depth questions that require separate commonsense knowledge is also difficult. To overcome these limitations, the present study proposes a method of generating a scene graph, which comprises explicit scene knowledge, from the images and using the graph in answering questions. This method can verify the validity of the answer through scene knowledge that can be understood by humans. Furthermore, it is capable of answering questions that require separate commonsense knowledge because it can combine scene knowledge with an external knowledge base.

2.3. Knowledge-Based VQA. Unlike the conventional VQA, Wang et al. [28] proposed a fact-based VQA (FVQA) problem that always requires separate commonsense knowledge to answer questions. To solve an FVQA problem, the image understanding result and external knowledge base must be used together to answer a question.

Wang et al. [28] tried to use only the knowledge that is directly related to the image from the knowledge defined in the external knowledge base. To that end, they obtained commonsense knowledge related to the image by searching the result of image understanding by using a deep neural network in the external knowledge base. In addition, they reasoned the answer to a question based on the obtained commonsense knowledge. However, it was difficult to answer questions that only required simple image understanding because the model could only generate answers based on commonsense knowledge. To overcome this limitation, Narasimhan and Schwing [29] proposed a VQA model which decides between scene and external knowledge to answer a question by applying the recurrent neural network to the question. However, this method is limited in that it only depends on one of the two knowledge types and cannot use a combination of the two knowledges. To obtain an answer by using both knowledge types simultaneously, Narasimhan et al. [30] combined the knowledge types into one graph and then extracted features from the graph by using the graph convolutional network and generated an answer by using the extracted features.

However, the previous studies performed only low-level image understanding to generate scene knowledge and are not appropriate for question answering problems that must consider relationships between objects, for example, "What things are in the refrigerator?" Furthermore, they only selected a simple recognition result or one of the known factors through an external knowledge base to answer a question. Therefore, the existing models can only answer

questions that ask simple facts including object and place, such as "What is the plant-eating animal shown here?" and cannot answer questions that require complex reasoning such as "How many fruits are on the kitchen table?" To overcome this limitation, the present study generates a 3D scene graph composed of objects in the environment, attributes, and spatial relationships between objects. In addition, to answer questions that require complex reasoning, the proposed method expresses the scene graph and external knowledge base based on ontology and generates an answer that considers both knowledge types together through knowledge reasoning.

3. Visual Experience-Based Question Answering

3.1. Problem Description. The proposed VEQA is a VQA problem about the agent's visual experience in a 3D simulated environment. The VEQA problem has several assumptions different from the conventional VQA problems. The given environment is only partially observable according to the visible range of the agent, and the environmental state can be changed by executing the agent's actions. Furthermore, the environmental change caused by the agent's actions may have some uncertainty. The natural language questions are limited to the visual experience of the agent.

The VEQA problem is expressed by $S = (A, Q, W, M, K)$, where A is a set of actions a_i to be taken by an agent in a photo-realistic 3D simulated environment, Q is a set of natural language questions q_j, W is a set of answers w_k, M is a set of observed RGB-D images I_l, and K is a set of scene graphs G_t that represent the estimated environmental states based on the agent's visual experience history $H_t = \langle (I_0, a_0), \ldots, (I_{t-1}, a_{t-1}), (I_t) \rangle$ until time t. The goal of the VEQA problem is to make an answer $w_t \in W$ to the given question $q_t \in Q$ based on the scene graph $G_t \in K$.

Specifically, the agent can take one of twelve different actions in the 3D simulated environment: eight agent pose-changing actions of "Move Ahead/Back/Left/Right," "Rotate Left/Right," and "Look Up/Down," and four interaction actions of "Open/Close Object," "Pick Up Object," and "Put Down Object." Among these actions, for the object interaction actions, the target object class is provided together with the action type, such as "Pick up Bread," as shown in Figure 1(b). The natural language questions are largely classified into six types: questions asking the existence of a specific object "Is there a potato somewhere in the room?", the number of a specific object "How many apples are there?", the attribute of a specific object "What is the color of plate?", the relationship between two objects "What is the relationship between pan and bread?" and the objects included in certain objects "What things are in the refrigerator?", and the owned object of the agent "What is the thing the agent has?".

3.2. Scene Graph. A scene graph G_t in the VEQA problem structurally represents an environmental state s_t estimated based on the agent's visual experience history $H_t = \langle (I_0, a_0), \ldots, (I_{t-1}, a_{t-1}), (I_t) \rangle$. A 3D scene graph, $G_t = (N_t, E_t)$, is composed of a set of nodes, N_t, and a set of edges, E_t. In a scene graph G_t, each node $n \in N_t$ represents an object or an attribute value that the object can have at time t. Consequently, the node set N_t is expressed as $N_t = O_t \bigcup A_t$, where O_t denotes a set of objects in the environment and A_t denotes a set of possible attribute values. For example, $O_t = \{\text{Apple}, \text{Mug}, \text{Chair}, \ldots,\}$ and $A_t = \{\text{Red}, \text{Gray}, \text{Opened}, \text{Closed}, \ldots,\}$. In this study, each object has two different attributes representing its color and openness. The color attribute of an object can have one of the predefined six color values, such as "Red" and "Gray," and the openness attribute of an object, such as a refrigerator and a dresser, can have one of the three values, "Opened," "Closed," and "Unable."

Furthermore, an edge $e \in E_t$ in a scene graph G_t represents a 3D spatial relationship between two objects at time t or a specific attribute of one object. Therefore, the edge set E_t is expressed as $E_t = R_t \bigcup C$, where R_t denotes a set of spatial relationships between two objects and C denotes a set of predefined attributes of each object class. For example, $R_t = \{\text{Right\_Of}, \text{Behind\_Of}, \text{Over}, \text{In}, \ldots,\}$ and $C = \{\text{Color}, \text{Openess}\}$. In this study, we define 9 different spatial relationship types between 25 different object classes to represent a 3D scene graph: "Left_Of," "Right_Of," "InFront_Of," "Behind_Of," "Over," "Under," "In," "On," and "Has."

3.3. Data Collection. The VEQA dataset was collected using the 3D indoor virtual environment, AI2-THOR [31]. To collect the VEQA dataset, we first defined 200 action scenarios that include a series of agent actions to be executed in different initial configurations of the environment. Each action scenario contains approximately 77 actions on average. Then, we collect input image and the corresponding scene graph data by executing each predefined action scenario using the simulated environment AI2-THOR. The question-answer data was generated semiautomatically based on scene graphs per every 10 agent actions in each action scenario. Consequently, a total 3,916 scene graphs including 13,109 objects, 26,218 attributes, and 25,583 relationships were built, and 5,397 question-answer pairs of six different types were generated. Table 1 lists the detailed specifications of this VEQA dataset. 80% of the VEQA dataset was used as the training set, 10% as the validation set, and 10% as the test set, respectively.

4. System Design

4.1. System Overview. To solve the abovementioned VEQA problem, we propose the VEQA system (VQAS), as shown in Figure 3. The proposed VEQA system first generates the scene graph representing the current environmental state based on observed images and agent actions and then makes a correct answer to the given question through knowledge reasoning on the scene graph. Accordingly, the VQAS

Table 1: Specification of the VEQA dataset.

Category		Count
Action scenario	Action scenarios	200
	Actions per action scenario	77
Question	Existence	1,168
	Counting	1,168
	Attribute	1,168
	Relation	1,005
	Include	676
	AgentHas	212
	Total questions	5,397
	Vocabulary size	90
Scene graph	Scene graphs	3,916
	Objects	13,109
	Attributes	26,218
	Relationships	25,583

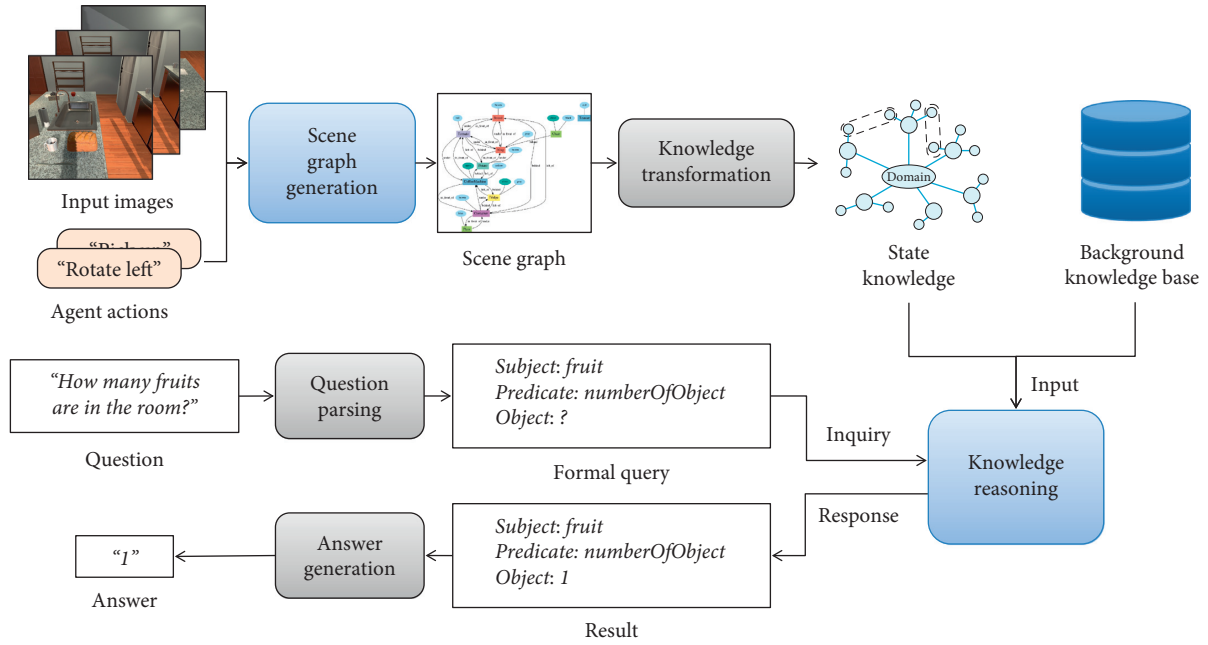

Figure 3: Visual experience-based question answering system.

consists of two subsystems: a scene graph generation system and a knowledge reasoning system.

The scene graph generation system uses a novel 3D scene generation model which can generate a series of 3D scene graphs from successive partial input images in a dynamic environment. The model consists of two different modules: a state recognition module and a state prediction module. The state recognition module expands the previous scene graph into the updated current one by considering the new partial image. On the other hand, the state prediction module predicts the current scene graph by applying effects of the executed action into the previous scene graph. The generated scene graphs are transformed into the formal state knowledge representation for use in knowledge reasoning.

In the knowledge reasoning system, a correct answer to the question is derived by applying predefined reasoning rules over a set of facts. These facts are from the static background knowledge on the environment as well as the dynamic state knowledge generated on the fly by the scene graph generation. The state knowledge corresponds to the environmental state representation, describing the current attributes and spatial relationships of objects. On the contrary, the background knowledge base corresponds to a set of facts that are predefined or assumed about the environment. A given natural language question is parsed to a formal query in triple form <subject, predicate, object> for knowledge reasoning using a deep neural network-based semantic parser. The knowledge reasoning system [32] is implemented on top of the SWI-Prolog RDF/OWL inference engine to derive the answer to the query over the abovementioned state and background knowledge facts.

The proposed VEQA system (VQAS) uses a predefined ontology for knowledge representation based on Description Logic (DL). The ontology is composed of various

hierarchical classes and their properties, as shown in Figure 4. To express state knowledge and background knowledge base, all the objects types that can appear in the environment, object attributes, and relationships between objects must be predefined using the classes and properties in the ontology. For example, through the IS-A relationship between the "Apple" and "Fruit" classes defined in the ontology, as shown in Figure 4, the prior background knowledge that "Apple" is a type of "fruit" can be used in the question. Furthermore, the state knowledge is expressed as instances of ontology. Each instance of ontology is expressed in three parts: the subject, predicate, and object.

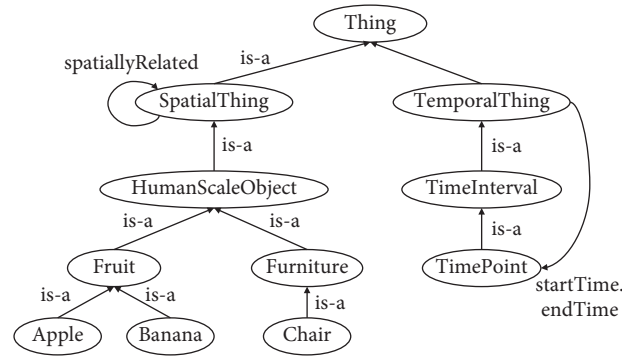

Figure 4: Example part of ontology.

4.2. Scene Graph Generation. To generate a scene graph of the entire environment from a partially observed image, an overall visual understanding of multiple images taken from different viewpoints in the environment is required. Furthermore, the scene graph must be updated and expanded in real time whenever a new image is observed in the VEAQ environment. However, the simultaneously processing of the increasing number of images is expensive and complex.

To solve this problem, we design a scene graph generation model to expand the previous scene graph G_{t-1} into the current scene graph G_t by considering a new image I_t, as described in the following equations:

$$G_t' = g_1(I_t), \tag{1}$$

$$G_t = f(I_t, G_{t-1}) = f(G_t', G_{t-1}). \tag{2}$$

Our model also considers dynamics of the environment to generate accurate 3D scene graphs. We can get more accurate scene graphs by considering environmental changes caused by the agent's actions additionally. This can complement wrong recognition to construct a partial scene graph G_t' from the current image I_t. As shown in the following equations, we get the predicted current scene graph $\hat{G}_t$ representing the resulting state after executing the action a_{t-1} by applying the corresponding action model into the previous scene graph G_{t-1}. And then, based on the previous scene graph G_{t-1}, the recognition-based scene graph G_t' and the prediction-based scene graph G_t'' are combined into the current scene graph G_t.

$$G_t'' = g_2(a_{t-1}, G_{t-1}), \tag{3}$$

$$G_t = f(G_t', G_t'', G_{t-1}). \tag{4}$$

Figure 5 shows the process of generating a scene graph G_t at time t. While the state recognition module in this figure corresponds to the function g_1 used in equation (1), the state prediction module plays the same role of the function g_2 used in equation (3). Based on the trained deep neural network, the state recognition module builds a partial scene graph G_t' from a new input image without any consideration of the executed agent action. However, there can be some errors in the partial scene graph G_t' due to challenging visual recognition. On the other hand, the state prediction module constructs the estimated current scene graph G_t'' by predicting only the environmental changes caused by the last executed action a_{t-1} without any consideration of the newly observed image I_t. Therefore, there can be some errors in the estimated scene graph G_t'' due to wrong action models. To complement the weakness of two modules, the proposed scene graph generation model builds the current scene graph G_t by combining the recognition-based partial scene graph G_t' with the prediction-based scene graph G_t'', as formulated in equation (4).

The state recognition module to extract a partial scene graph from a new image consists of four different neural networks: Object Detection Network (ODN), Three-dimensional Localization Network (TLN), Attribute Recognition Network (ARN), and Relationship Recognition Network (RRN), as shown in Figure 6.

First, to recognize object O_t' observed in image I_t, 2D object detection and 3D localization were performed sequentially. In 2D object detection, the 2D bounding boxes of objects in the image were determined using Object Detection Network (ODN) based on YOLOv3 [33], which can detect even small objects. After 2D object detection, the 3D bounding boxes of detected objects are determined through 3D Localization Network (TLN). In many conventional 3D object detection models, the object positions are expressed in relative coordinates centered on the viewpoint of agent as an observer [34]. However, to construct an invariant 3D scene graph regardless of the agent's position, the object positions should be expressed in absolute coordinates centered on a specific point in the environment.

We design a 3D Localization Network (TLN) as shown in Figure 7, considering this problem. The TLN first extracts the relative position feature from the 2D bounding box of an object and the depth image [34]. This feature involves relative position information of the object based on agent pose. And then, the TLN predicts the origin of absolute coordinates by using both relative positions of the object and the agent. To improve the prediction accuracy, the TLN network also makes use of the object class information extracted by the Object Detection Network (ODN).

After 3D localization of objects detected in the image, the attributes of individual objects, A_t', are recognized through the Attribute Recognition Network (ARN), as shown in Figure 8. The ARN determines one of the predefined values for some object attributes such as color and openness. These

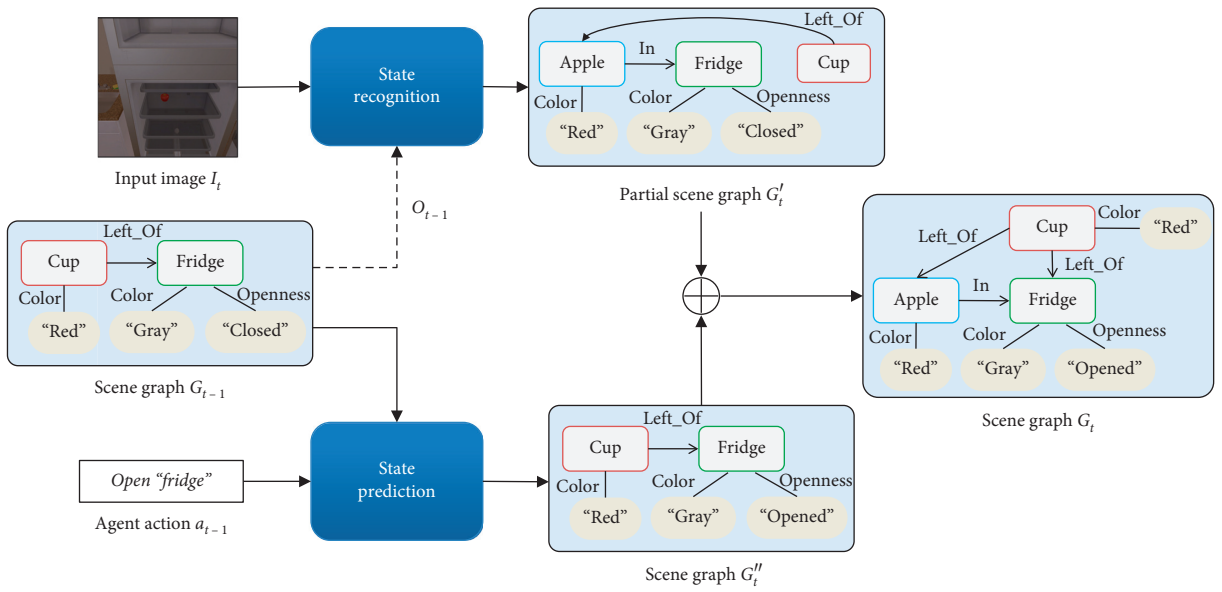

Figure 5: Overall process of scene graph generation.

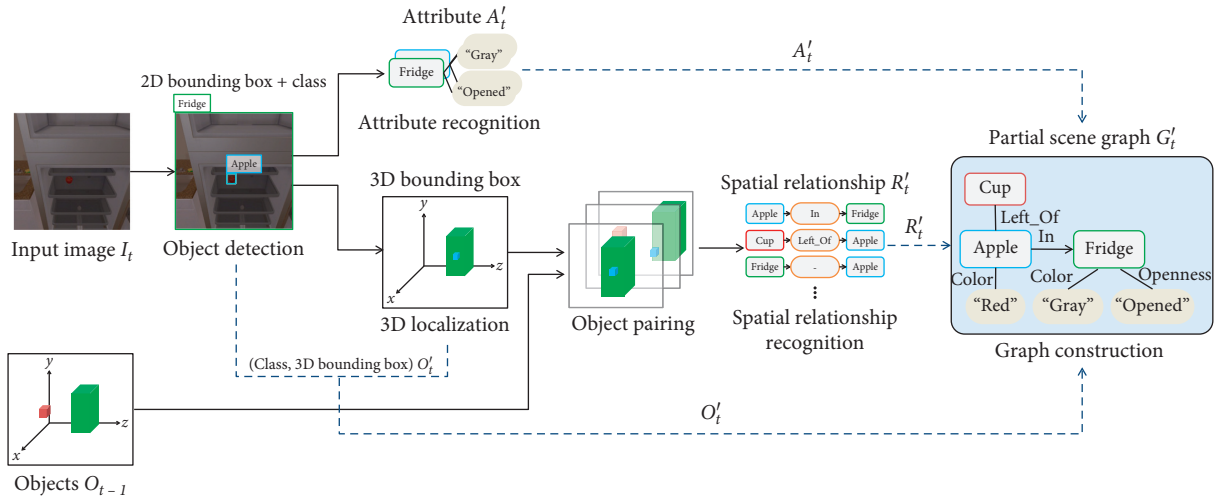

Figure 6: Partial scene graph generation.

object attributes can be recognized depending on visual features extracted from the image through a convolutional neural network (CNN) like ResNet or VGG. However, we design the ARN network to make use of the object class information additionally. The class information of an object often helps to estimate a certain attribute of the object. For example, we can predict the color of an apple that must be red or green without visual information.

Finally, to detect all possible relationships R'_t between objects, the state recognition module generates a set of object pairs between the detected objects. An object pair is composed of a subjective object and an objective object. As the VQAS system assumes a partially observable environment, the relationships between two objects detected independently in different images as well as the relationships between two objects detected in the same image should be recognized all together. Consequently, for relationship recognition at time t, all possible object pairs are generated using the objects O'_t detected in the image I_t at time t and the objects O_{t-1} already included in the previous scene graph G_{t-1}.

After all possible object pairs are generated, the spatial relationship matching to each object pair is determined through the Relationship Recognition Network (RRN), as shown in Figure 9. The RRN receives the 3D bounding boxes of two objects as an input to recognize the spatial relationship between them. It also receives the class information of two objects as another input because the allowed relationships between two objects often depend on the classes of participating objects. For example, "Apple" and "Spoon"

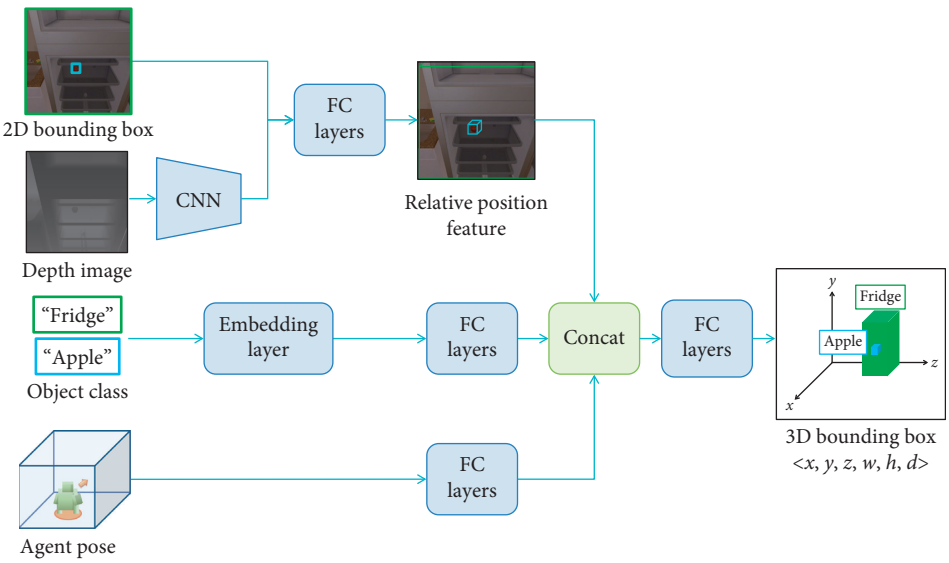

Figure 7: Three-dimensional Localization Network (TLN).

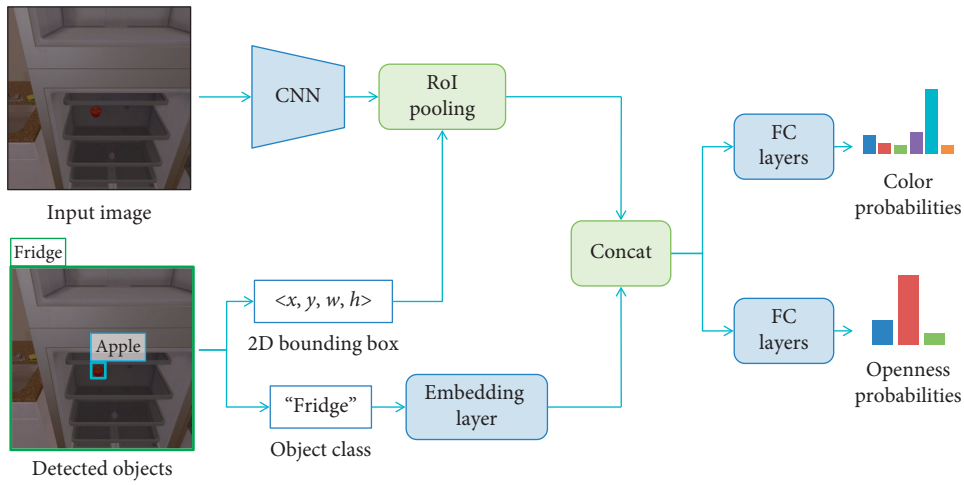

Figure 8: Attribute Recognition Network (ARN).

cannot have the "In" relationship, but "Apple" and "Fridge" can.

As mentioned above, the state recognition module can only recognize the environment state s_t using a single image I_t and cannot consider the environmental changes that do not appear in the image I_t. For example, in Figure 1(b), when the position of "Bread" and its relationship with another object are changed through action a_0 = "Pick up Bread," it is difficult for the state recognition module to detect this environmental change. Besides, there can be some errors in the resulting partial scene graph G_t' due to challenging visual recognition. To overcome these problems, the proposed scene graph generation model also makes use of the action-based state prediction module.

For state prediction, the expected effect of environmental change is defined for each agent action as an action model. Figure 10 shows some examples of the action models represented in Planning Domain Description Language (PDDL) [35]. As the VQAS system assumes an environment with uncertainty in the actions, the potential effect of environmental change is represented with the actual occurrence probability, as shown in Figure 10. In the state prediction module, the effect of environmental change caused by the last action a_{t-1} is obtained first from the corresponding action model. Then, the effect is applied to the previous scene graph G_{t-1} to generate the scene graph G_t''.

For more accurate scene graph generation, the proposed system combines two scene graphs G_t' and G_t'' generated through state recognition and state prediction in a complementary manner. Before the combination, the objects in the two scene graphs were compared to determine whether they are the same object. To that end, the 3D intersection over union (IoU) was determined, which indicates the degree of intersection of two objects in a 3D space. If the 3D

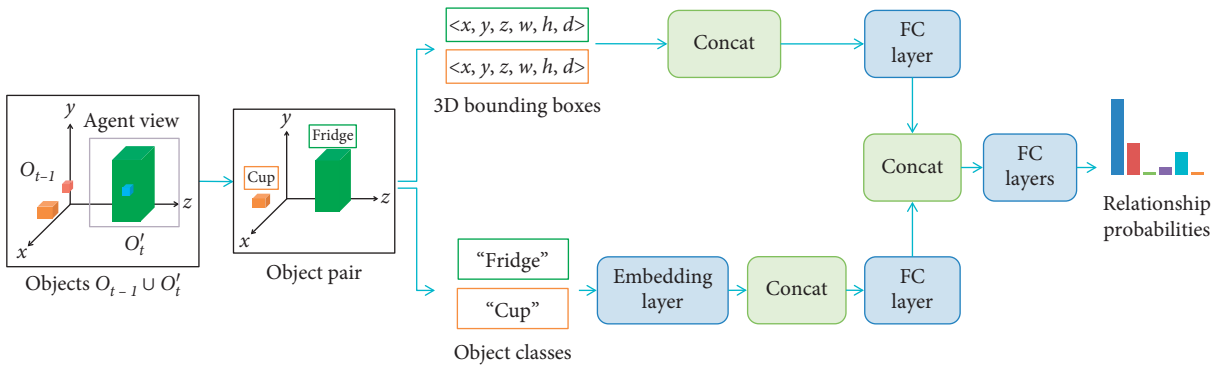

Figure 9: Relationship Recognition Network (RRN).

```
(:action open-close-object
 :parameters (?obj – Object ?init-att – Openness ?goal-att – Openness)
 :effect (probabilistic 0.8 (and (not (openness ?obj ?init-att))
                                 (openness ?obj ?goal-att))))

(:action pick-up-object
 :parameters (?target-obj – Object ?support-obj – Object ?agent – Agent
              ?pose-target-obj – Pose ?pose-agent – Pose)
 :effect (probabilistic 0.8 (and (has ?agent ?target-obj)
                                 (not (in ?target-obj ?support-obj))
                                 (not (on ?target-obj ?support-obj))
                                 (not (at ?target-obj ?pose-target-obj))
                                 (at ?target-obj ?pose-agent))))

(:action put-down-object
 :parameters (?target-obj – Object ?support-obj – Object ?agent – Agent
              ?pose-agent – Pose ?pose-to – Pose)
 :effect (probabilistic 0.8 (and (probabilistic 0.9 (in ?target-obj ?support-obj)
                                                0.1 (on ?target-obj ?support-obj))
                                 (not (has ?agent ?target-obj))
                                 (not (at ?target-obj ?pose-agent))
                                 (at ?target-obj ?pose-to))))
```

Figure 10: Some examples of the action model.

IoU is greater than 0.3, the two objects are determined as identical. After all the objects were compared in this way, the objects that only existed in one of the two scene graphs were added to scene graph G_t with attributes and relationships of the objects. In contrast, the objects that were included in both scene graphs were added to scene graph G_t in their original forms only if the object attributes and relationships were the same in both scene graphs. If the object attributes or relationships differed, they were integrated as follows and the result was added to scene graph G_t. Two different results are combined based on the probability distribution of each class. In the case of state recognition, the probability distribution of each class obtained from the recognition network is used. In the case of state prediction, the probabilities that appear in the environmental change effect were used in probability distribution. The weighted average of these probabilities was determined for each class, and the highest value was selected as the final result. The following equation expresses this integration method:

$$\text{Fusion}(d', d'', r) = \arg\max(d' * r + d'' * (1 - r)), \quad (5)$$

where d' and d'' are the probability distribution of classes obtained through recognition and prediction, respectively, and r is the weighted ratio of the recognition result. The higher the recognition accuracy is, the more advantageous it is to have a higher value. In this study, a higher weight was given to the state recognition result. The weighted average method has the advantage of always providing a result that complements the recognition and prediction results. However, its disadvantage is that if the accuracy of the recognition or prediction result is low, the combined result also has a low accuracy.

The generated scene graphs are transformed into the formal state knowledge representation for use in knowledge reasoning. A knowledge fact representing the environmental state is expressed as a triple of <subject, predicate, object>, and a state knowledge k_t is a set of these facts. On the other hand, a scene graph G_t may include multiple relationship/attribute edges, each of which corresponds to a knowledge fact. Thus, the proposed system transforms each relationship/attribute edge with two participating objects into a knowledge fact expressed as a triple of <subject, predicate, object>. Figure 11 illustrates the transformation of a scene graph G_t into a state knowledge k_t based on the predefined ontology.

4.3. Knowledge Reasoning for QA. The knowledge reasoning is one of the techniques that have been mainly used in traditional artificial intelligence for a long time. This technique can be used to find new facts from known facts or to correctly deduce answers corresponding to a given question, based on predefined reasoning rules [32]. This technique has many advantages such as explainable inference process and simple incorporation of large prebuilt background knowledge. To take these advantages, we use a knowledge reasoning system for scene graph-based question answering in our VQAS system.

This section describes the knowledge reasoning process for visual experience-based question answering (VEQA) with the state knowledge and a background knowledge base, as shown in Figure 12. In the knowledge reasoning process, both the state and the background knowledge are assumed to be built based on the same context ontology. When a natural language question about an environmental state is given, it is first transformed to a formal query for knowledge reasoning. Then, an answer to the question is derived by performing

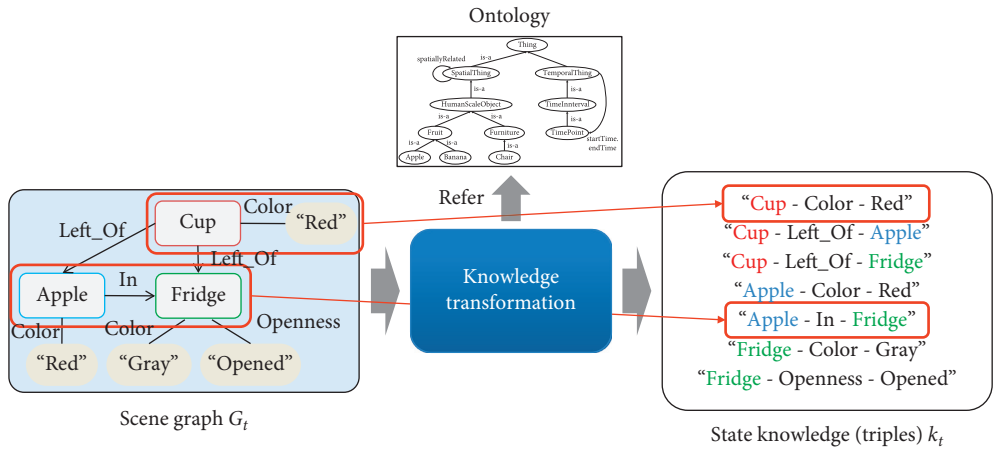

Figure 11: Example of knowledge transformation.

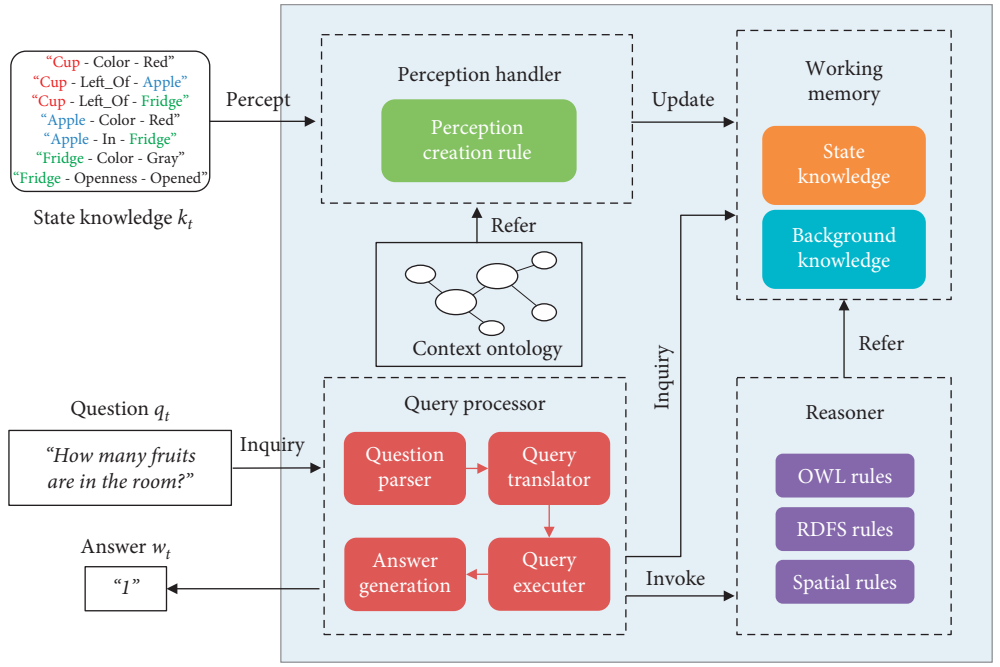

Figure 12: Knowledge reasoning for question answering.

knowledge reasoning over the knowledge source combining the corresponding state knowledge and the background knowledge, starting from the formal query.

Figure 12 shows the knowledge reasoning process for visual experience-based question answering (VEQA). The state knowledge k_t is generated in real time as the agent performs an action a_{t-1} and receives an input image I_t. Each time the state knowledge k_t is generated, the perception handler stores and updates it in the working memory. Furthermore, the background knowledge B is loaded from a background-knowledge base when the system is started. However, unlike the state knowledge k_t, the background knowledge B remains constant in the working memory from the system initiation because it does not change in the given environment. Two different knowledge types stored in the working memory are used as the knowledge source for knowledge reasoning to answer a given question.

For knowledge reasoning, a natural language question is parsed into a formal query by a trained Question Parsing Network (QPN). A formal query is expressed as a triple of <subject, predicate, object> representing a natural language question. For example, the natural language question "How many fruits are in the room?" in Figure 12 can be changed to the triple query, <Fruit, NumberOfObject, ?>.

The proposed system transforms a natural language question into a triple query by using a Question Parsing Network (QPN), as shown in Figure 13. The expression of natural language questions is characterized by a sequence of words. Consequently, the features of natural language questions are extracted by the bidirectional long short-term

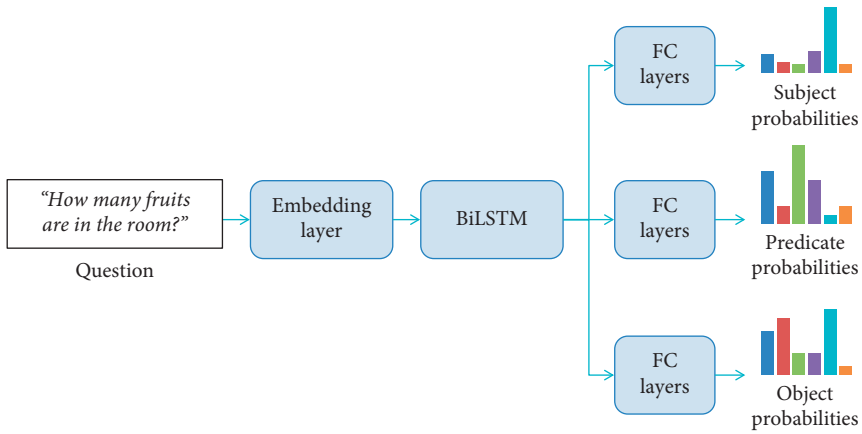

Figure 13: Question Parsing Network (QPN).

memory of a recurrent neural network, which can process sequence data. Furthermore, the subject, predicate, and object comprising triple queries are predicted through the extracted features. This QPN is used as the question parser of the query processor in Figure 12.

In query processing, the natural language question is first transformed into a through-the-question parser. Then, the triple query is expressed as a SWI-Prolog [36] query according to the predefined reasoner by using the query translator. Next, the question executer infers the answer to the question by querying the working memory or by using the reasoner. The reasoner generates new knowledge based on the knowledge types built in the working memory by using predefined reasoning rules. Figure 14 illustrates the reasoning rules defined in SWI-Prolog; these comprise other predefined predicates, each of which obtains a fact satisfying the condition by querying the triple knowledge in the working memory and class layer of the ontology. These reasoning rules are defined one by one for each query type so that the correct answer can be generated for every given query. Lastly, the knowledge obtained through the reasoning rules is transformed from the SWI-Prolog format into natural language format that can be understood by humans through the answer generator.

```
existenceOfObject(ObjectType, Object):-
        atom_concat(,ObjectType, X),
        rdf(Object, rdf:type, X).

numberOfObject(ObjectType, Count):-
        atom_concat(ObjectType, X),
        findall(Object, rdf(Object, rdf:type, X),Objects),
        count(Objects, Count).

mainColorOfObject(Object, Color):-
        atom_concat(Object, X),
        rdf(Object_instance, rdf:type, X),
        rdf(Object_instance, owl:mainColorOfObject, Color).

relationOfTwoObjects(Subject, Object, Relation):-
        atom_concat(Subject, Sub_Class),
        atom_concat(Object, Obj_Class),
        rdf(Sub_Instance, rdf:type, Sub_Class),
        rdf(Obj_Instance, rdf:type, Obj_Class),
        rdf(Sub_Instance, Relation, Obj_Instance).
```

Figure 14: Example of reasoning rule.

5. Implementation and Evaluation

5.1. Model Training. To implement the proposed system, deep neural network modules comprising the system were trained independently. These modules to be trained included TLN for 3D scene graph generation, ARN, RRN, and QPN for question answering. First, as the loss function for training, the TLN used the mean abstract error, as follows:

$$\text{MAE} = -\sum \left| b_i - \hat{b}_i \right|, \tag{6}$$

where b and $\hat{b}$ denote the predicted and accurate 3D bounding boxes, respectively. For the other deep neural network modules, the following cross entropy error was used:

$$\text{Cross Entropy Error} = -\sum y_i \log \hat{y}_i, \tag{7}$$

where y denotes one-hot encoding value for the ground truth answer class and $\hat{y}$ denotes the probability for each class predicted by the model.

Furthermore, each network was trained using Adam optimizer with a learning rate of 0.001. Also, we used the step decay that decreases the learning rate 10 times per each 10 epochs. The proposed system receives a 2D bounding box of an object that includes a small error because the TLN and ARN use the recognized results from the 2D object detection network (ODN). Therefore, in this study, random noise, considering the error range of the ground truth 2D bounding box, was added when training the TLN and ARN. In addition, each neural network was implemented with PyTorch, a Python deep learning library, and trained with GeForce GTX TITAN X GPU.

6. Experiments

To evaluate the performance of the proposed VQAS system, we conduct several experiments with the VEQA benchmark dataset. (1) The first experiment is performed for evaluating the question answering performance of the proposed system. In this experiment, different VQAS system configurations with ground truth data are compared with each other: VQAS (without any ground truth data), VQAS with ground truth 2D objects, VQAS with ground truth scene graphs, VQAS with ground truth queries. Table 2 shows the experimental results classified into six different types of VEQA questions. The experimental results showed that the proposed VQAS system without any ground truth data achieved a high performance of 72.37% in average for all types of questions. The system performance when the ground truth class and 2D bounding box of the objects are directly given (VQAS with GT 2D objects) shows a difference of 11% with VQAS. This shows that the 2D object detection performance significantly affects the visual experience-based question answering performance because the object detection result affects attribute recognition, 3D localization, and relationship recognition. Next, the system performance when the ground truth scene graph was used for question answering (VQSA with GT scene graphs) shows a result of close to 100%, implying that the VEQA problem can be solved sufficiently through correct scene graph generation.

Furthermore, the experimental result when the ground truth accurate query was used without using the QPN (VQAS with GT query) shows that the performance result has no significant difference from that of the original VQAS. This suggests that the QPN can predict the corresponding correct query from the natural language questions. Furthermore, the overall result shows that the performances of existence and counting type questions are the highest because the proposed system can explicitly represent objects in the environment and infer the accurate answer based on explicit knowledge. The question type that showed the lowest performance was *Include*. To answer an *Include* question in the proposed system, correct object detection and relationship recognition are required. Furthermore, all objects with an *In* relationship with a specific object must be correctly recognized. Consequently, the *Include* question type showed a lower performance than other question types.

(2) The second experiment is performed for evaluating the scene graph generation performance of the proposed VQAS system with different state estimation methods. In this experiment, two different state estimation methods are compared: VQAS without SP (without state prediction) and VQAS (with action model-based state prediction). However, both two methods share the same state recognition module to estimate the current environmental state. Two different performance measures are used for this experiment: Object mAP for 3D object detection and SGGen for scene graph generation. Object mAP represents the mAP (mean Average Precision) of objects, the class of which is the same as that of the ground truth object, and for which the 3D IoU (Intersection over Union) is greater than 0.3. The SGGen represents the recall of the triples comprising the ground truth scene graph. Furthermore, the experiments were performed separately according to the attribute and object relationship depending on the predicate type of the triple. Each triple was determined as the ground truth answer when the object class of the triple was the same as that of the ground truth object and has a 3D IoU of more than 0.3, and for which the other relationship and attribute classes matched those of the ground truth.

The experimental result in Table 3 confirmed that the use of the action model-based state prediction helps improve the performance of 3D object detection and that of scene graph generation. Regarding object detection, the action model allows the accurate prediction of the object's positional changes. For example, when the agent picks up an object or moves while holding an object, it is difficult to determine the positional changes of the object in the images. In contrast, the moved position of the object can be predicted through the action model. For scene graph generation, both the performances for attributes and relationships improved. This suggests that the state prediction result using action models and the state recognition result generated through visual recognition can be combined in a complementary manner.

(3) The third experiment is performed for evaluating the scene graph generation performance of the proposed VQAS system with different state recognition models. In this experiment, four different state recognition models are compared: $(O + A + T + R)$, $(O + A + T_{gt} + R)$, $(O + A_{gt} + T_{gt} + R)$, and $(O_{gt} + A_{gt} + T_{gt} + R)$. O, A, T, and R represent the cases of using deep neural network modules for 2D object detection, attribute recognition, 3D localization, and relationship recognition. In contrast, O_{gt}, A_{gt}, T_{gt}, and R_{gt} represent the cases of using the ground truth instead of a deep neural network module. Table 4 shows the results of this experiment. In Table 4, the proposed state recognition model $(O + A + T + R)$ shows an SGGen performance of 53.73% and can thus generate scene graphs above a certain level of accuracy. However, the more the recognition networks are used, the lower the performance is. This seems to be because the recognition error in each recognition network has a negative effect on the recognition of the next recognition network. In particular, when the result of the system using the object detection network $(O + A_{gt} + T_{gt} + R)$ is compared with the result of the system using ground truth $(O_{gt} + A_{gt} + T_{gt} + R)$, both mAP and SGGen decrease when the objects are automatically detected in the image by using an object detection network. This is because the performance of the object detection network affects all the other recognition networks using the result. In other words, if the objects in the input image cannot be found accurately, the accuracy of the scene graph representing the object attributes and their spatial relationships will inevitably decrease.

Table 2: Performance analysis of visual experience-based question answering depending on different VQAS configurations.

Configurations	Accuracy (%)						
	Question types						Total
	Existence	Counting	Attribute	Relationship	Include	AgentHas	
VQAS	91.96	79.60	68.53	61.16	56.24	63.21	72.37
VQAS with GT 2D objects	99.74	91.18	78.91	73.94	72.93	78.30	83.37
VQAS with GT scene graph	99.74	100.0	99.91	93.23	99.70	100.0	98.62
VQAS with GT query	92.22	79.60	68.53	64.18	56.24	63.21	72.95

Table 3: Performance analysis of scene graph generation depending on different state estimation methods.

State estimation methods	Object mAP (%)	SGGen (%)		
		Attribute	Relation	Total
VQAS without SP	63.32	56.62	40.99	50.11
VQAS	**63.91**	**58.69**	**41.48**	**51.26**

Bold values represent the best results.

Table 4: Performance analysis of scene graph generation depending on different state recognition models.

Models	Object mAP (%)	SGGen (%)		
		Attribute	Relation	Total
$O + A + T + R$	68.79	56.87	46.79	53.73
$O + A + T_{gt} + R$	85.12	69.69	60.09	67.35
$O + A_{gt} + T_{gt} + R$	85.12	84.61	60.09	79.62
$O_{gt} + A_{gt} + T_{gt} + R$	100.0	100.0	95.89	98.80

(4) The fourth experiment is performed for evaluating 3D localization performance of the TLN neural network with different input information. In this experiment, three different input pieces of information for the TLN neural network are compared: (Depth Image + 2D Bbox), (Depth Image + 2D Bbox + Agent Pose), and (Depth Image + 2D Bbox + Agent Pose + Object Class). Table 5 shows the results for this experiment. The results in the *positions* column indicate <*x*, *y*, *z*> corresponding to a position in the predicted 3D bounding box, and the *size* indicates <*w*, *h*, *d*> corresponding to size. Furthermore, the mean abstract error, which is the difference from the ground truth, was used as a measure. The *accuracy* indicates the ratio of 3D IoU values greater than 0.3 from the ground truth. The result in Table 5 suggests that the performance is very low when only the depth image and 2D bounding box were used, because both input pieces of information represent only the relative position of the object from the agent's perspective. In contrast, when the agent pose information was used together with Depth Image + 2D Bbox, the performance is seen to greatly improve compared to the result of the system that did not use the agent pose. This is because the agent pose provides information that can reveal the absolution position from the relative position of the object. Furthermore, the performance improved further when the object class was used (Depth Image + 2D Bbox + Agent Pose + Object Class). This is because the object class shows the general size of the object. For example, the size of "Apple" is small, and the size of "Fridge" is large.

(5) The fifth experiment is performed for evaluating attribute recognition performance of the ARN neural network with different input information. In this experiment, three different input pieces of information for the ARN neural network are compared: (Image + 2D Bbox), (Object Class), and (Image + 2D Bbox + Object Class). Table 6 shows the results for this experiment. First, when only the image and the 2D bounding boxes of objects were used for attribute recognition (Image + 2D Bbox), both color and openness showed performances higher than 80%. In contrast, when only the object class was used for attribute recognition (Object Class), only the performance of the openness was high. Similar to the experimental results in Table 5, the object class can be used for predicting the general value of each object class. For example, we can see that "Apple" has an attribute showing that it cannot be opened or closed, whereas "Fridge" can have the "Opened" or "Closed" value. Furthermore, the highest performance is obtained when the image, 2D bounding box, and object class are used together for attribute recognition (Image + 2D Bbox + Object Class). This is because the above-explained advantages for each input information are helpful for improving the independent performance.

(6) The sixth experiment is performed for evaluating relationship recognition performance of the RRN neural network with different input information. In this experiment, three different input pieces of information for the RRN neural network are compared: (3D Bbox), (Object Class), and (3D Bbox + Object Class). Table 7 shows the results for this experiment. These results confirm that spatial relationships can be recognized even when only the 3D bounding boxes of two objects are used for relationship recognition (3D Bbox). This is because the spatial relationships between two objects can be estimated to some degree through comparison of the positions of the two objects in the space. Furthermore, when only the object class is used for relationship recognition (Object Class), the resulting

Table 5: Performance analysis of 3D localization depending on different input information.

Input information	Output	Mean absolute error		Accuracy (%)
Depth Image + 2D Bbox	Position	0.344	0.522	17.89
	Size	0.178		
Depth Image + 2D Bbox + Agent Pose	Position	0.109	0.249	62.52
	Size	0.140		
Depth Image + 2D Bbox + Agent Pose + Object Class	**Position**	**0.089**	**0.216**	**78.93**
	Size	**0.127**		

Bold values represent the best results.

Table 6: Performance analysis of attribute recognition depending on different input information.

Input information	Output	Precision (%)	Recall (%)	Accuracy (%)
Image + 2D Bbox	Color	89.23	79.76	87.52
	Openness	**82.24**	33.33	82.24
Object Class	Color	53.01	43.39	49.99
	Openness	72.03	71.89	92.42
Image + 2D Bbox + Object Class	**Color**	**89.71**	**80.55**	**89.87**
	Openness	**79.57**	**79.48**	**94.57**

Bold values represent the best results.

Table 7: Performance analysis of relationship recognition depending on different input information.

Input information	Precision (%)	Recall (%)	Accuracy (%)
3D Bbox	75.28	80.72	86.04
Object Class	31.82	58.24	67.53
3D Bbox + Object Class	**76.34**	**94.24**	**89.67**

Bold values represent the best results.

performance is lower than that when only using the 3D bounding box (3D Bbox). However, the performance result shows that spatial relationships can be recognized to some degree by only using the object class. This is because the possible spatial relationships can be limited according to the object class. For example, as shown, "Bread" and "Apple" have spatial relationships excluding "In" and "On"; the "In" relationship can be expected for "Egg" and "Fridge." When two different pieces of information (3D Bbox + Object Class) were used together, the highest performance was obtained, implying that two different pieces of information can help improve performance independently.

Finally, Figure 15 shows a sequence of examples for qualitative performance analysis of the proposed VQAS system. First, the figures on the left side show image I_t observed at time t before the agent performs an action, detected object O_t, and generated scene graph G_t. The figures on the right side show image I_{t+1} observed at time $t+1$ after the action performance, detected object O_{t+1}, and generated scene graph G_{t+1}.

Figure 15(a) illustrates the scene graphs G_t and G_{t+1} generated before and after executing "Rotate Right," an action in which the agent changes its pose. An observation of the scene graph G_{t+1} reveals that the relationship between "Microwave" and "Spoon" is recognized as "Right_Of." However, while "Microwave" is not observed in image I_{t+1}, "Spoon" is newly observed in image I_{t+1}. This confirms that the proposed VQAS system can generate scene graphs properly even in a partially observable environment.

Figures 15(b) and 15(c) illustrate the scene graphs generated before and after executing actions of "Pick Up" and "Open," in which the agent directly changes the environmental state. The scene graph G_{t+1} in Figure 15(b) shows the proper prediction of the "Has" relationship of "Agent" and "Bread." The scene graph G_{t+1} in Figure 15(c) confirms that the openness of "Fridge" was modified and the newly observed objects are correctly recognized after performing "Open" action.

However, in the case of Figure 15(d), a wrong scene graph is generated owing to the recognition error. In this example, the 3D positions of "Fridge" appearing in the two images were recognized differently between images I_t and I_{t+1}. The same object was misclassified as "Bowl" in image I_t and "Pan" in image I_{t+1}; this error caused the generation of a wrong scene graph. In this example, the two "Fridges" and both "Bowl" and "Pan" appear in scene graph G_{t+1}. Thus, the recognition error in each recognition network has a negative effect on the scene graph. To improve this problem, future studies should devise a method to improve the performance of each recognition network and correct the recognition error and thus the derived one.

The table in the right side in Figure 15 shows the answers generated through triple query and knowledge reasoning predicted from the natural language question at time $t+1$. Figures 15(a) and 15(b) show the correct prediction of the triple query and generation of a correct answer. However, Figure 15(c) shows the correct prediction of the triple query but the generation of a wrong answer due to erroneous scene graph generation. This shows that the proposed system can

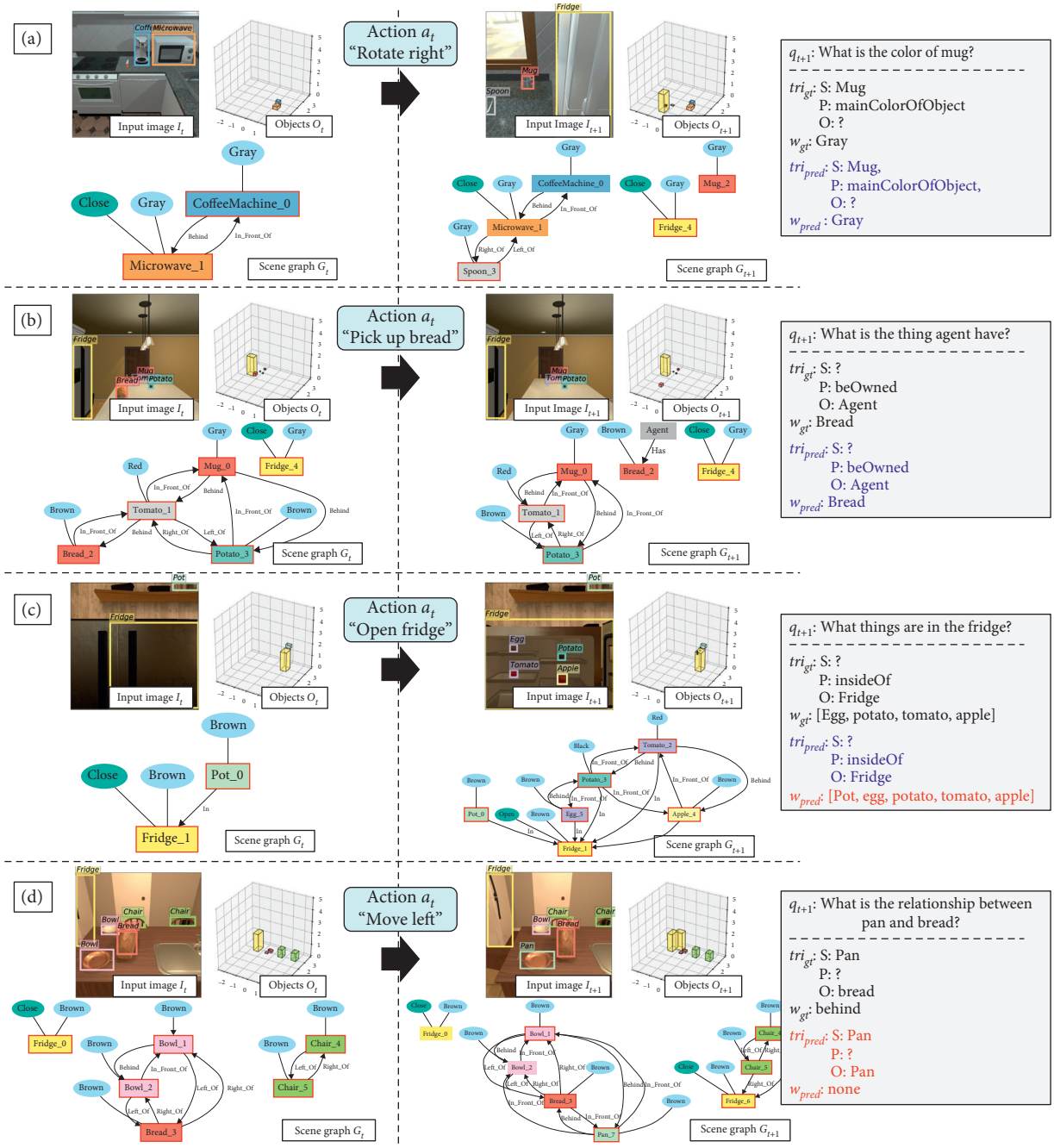

Figure 15: Examples of scene graph generation and question answering.

generate a wrong answer that appears in the scene graph, as in the case of Figure 15(c) because it performs knowledge reasoning based on state. Therefore, the proposed system requires the generation of a correct scene graph for answer generation. In the case of Figure 15(d), even though the correct scene graph was generated, the triple query was predicted incorrectly. This could be because reasoning was performed using a query with a different meaning from that of the given question. These examples suggest that the accuracy of the scene graph generation and the accuracy of question parsing both affect the answer generation.

7. Conclusion

This paper proposed a novel VEQA problem and the corresponding dataset for embodied intelligence research that requires an agent to do actions, understand entire 3D scenes from successive partial input images, and answer natural language questions about the dynamic scenes in a complex multimodal environment. To address the VEQA problem, we propose a hybrid visual question answering system, VQAS, integrating a deep neural network-based scene graph generation model and a rule-based knowledge reasoning

system. Furthermore, we propose a novel 3D scene graph generation model which can generate a series of 3D scene graphs from successive partial input images in a dynamic environment. The proposed model can overcome the limitation of the conventional scene graph generation models building just a 2D scene graph from a single still image. The model also meets well the partial observability and dynamics of the VEQA environment. We also propose a knowledge reasoning system to answer natural language questions based on 3D scene graphs. Different from the pure deep neural network-based models, the proposed knowledge reasoning system can use a rich knowledge source to answer questions by combining the shallow knowledge in 3D scene graphs with a large amount of prebuilt deep background knowledge.

In this study, a series of experiments using AI2-THOR and the VEQA benchmark dataset were performed to analyze the performance and limitation of the proposed VQAS system. The results of these experiments verified the usefulness of the VEQA problem and the high performance of the proposed VQAS system. However, a few limitations of the proposed system were also found, such as the possibility that the state knowledge may not express the observed environmental state perfectly or may generate a wrong answer when the estimated state is different from that of the actual environment. To improve these limitations, the generation of more accurate state knowledge will be researched in the future.

Data Availability

The VEQA data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Kyonggi University Research Grant 2019.

References

[1] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[2] D. Liu, M. Bober, and J. Kittler, "Visual semantic information pursuit: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, In press.

[3] S. Antol, A. Agrawal, J. Lu et al., "VQA: visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4–31, Santiago, Chile, December 2017.

[4] D. Gordon, A. Kembhhavi, M. Rastegari et al., "IQA: visual question answering in interactive environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4089–4098, Salt Lake City, UT, USA, June 2018.

[5] A. Das, S. Datta, G. Gkioxari et al., "Embodied question answering," in *Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition*, pp. 2054–2063, Salt Lake City, UT, USA, June 2018.

[6] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840, Salt Lake City, UT, USA, June 2018.

[7] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3076–3086, Honolulu, HI, USA, July 2017.

[8] D. Xu, Y. Zhu, C. B. Choy, and F. F. Li, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5410–5419, Honolulu, HI, USA, July 2017.

[9] Y. Li, W. Ouyang, B. Zhou, K. Wang, and Z. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1261–1270, Venice, Italy, October 2017.

[10] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, "Visual question answering: a survey of methods and datasets," *Computer Vision and Image Understanding*, vol. 163, pp. 21–40, 2017.

[11] D. Zhang, R. Cao, and S. Wu, "Information fusion in visual question answering: a survey," *Information Fusion*, vol. 52, pp. 268–280, 2019.

[12] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *Proceedings of the Computer Vision—ECCV 2016*, pp. 852–869, Amsterdam, The Netherlands, October 2016.

[13] M. Y. Yang, W. Liao, H. Ackermann, and B. Rosenhahn, "On support relations and semantic scene graphs," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 131, pp. 15–25, 2017.

[14] J. Johnson, R. Krishna, M. Stark et al., "Image retrieval using scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3668–3678, Boston, MA, USA, June 2015.

[15] N. Xu, A.-A. Liu, J. Liu, W. Nie, and Y. Su, "Scene graph captioner: image captioning based on structural visual representation," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 477–485, 2019.

[16] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1219–1228, Salt Lake City, UT, USA, June 2018.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 91–99, 2017.

[18] P. Gay, J. Stuart, and A. D. Bue, "Visual graphs from motion (VGfM): scene understanding with object geometry reasoning," in *Proceedings of the Asian Conference on Computer Vision*, pp. 330–346, Perth, WA, Australia, December 2018.

[19] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21–29, Las Vegas, NV, USA, June 2016.

[20] D. Yu, J. Fu, T. Mei, and Y. Rui, "Multi-level attention networks for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4709–4717, Honolulu, HI, USA, July 2017.

[21] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1367–1381, 2018.

[22] L. Gao, P. Zeng, J. Song et al., "Structured two-stream attention network for video question answering," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 6391–6398, 2019.

[23] L. Gao, P. Zeng, J. Song, and H. T. Shen, "Examine before you answer: multi-task learning with adaptive-attentions for multiple-choice VQA," in *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 1742–1750, Seoul, Korea, October 2018.

[24] J. Song, P. Zeng, L. Gao et al., "From pixels to objects: cubic visual attention for visual question answering," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 906–912, Stockholm, Sweden, July 2018.

[25] P. Anderson, X. He, C. Buehler et al., "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6077–6086, Salt Lake City, UT, USA, June 2018.

[26] D. Teney, L. Liu, and A. van den Hengel, "Graph-structured representations for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Honolulu, HI, USA, July 2017.

[27] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Learning to count objects in natural images for visual question answering," in *Proceedings of International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[28] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel, "FVQA: fact-based visual question answering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2413–2427, 2018.

[29] M. Narasimhan and A. G. Schwing, "Straight to the facts: learning knowledge base retrieval for factual visual question answering," in *Proceedings of the European Conference on Computer Vision*, pp. 451–468, Munich, Germany, September 2018.

[30] M. Narasimhan, S. Lazebnik, and A. G. Schwing, "Out of the box: reasoning with graph convolution nets for factual visual question answering," in *Proceedings of the Neural Information Processing Systems*, pp. 2654–2665, Montreal, Canada, December 2018.

[31] E. Kolve, R. Mottaghi, D. Gordon et al., "AI2-THOR: an interactive 3D environment for visual AI," 2017, https://arxiv.org/abs/1712.05474.

[32] S. Lee and I. Kim, "A robotic context query-processing framework based on spatio-temporal context ontology," *Sensors*, vol. 18, no. 10, p. 3336, 2018.

[33] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[34] M. M. Rahman, Y. Tan, J. Xue, L. Shao, and K. Lu, "3D object detection: learning 3D bounding boxes from scaled Down 2D bounding boxes in RGB-D images," *Information Sciences*, vol. 476, pp. 147–158, 2019.

[35] M. Fox and D. Long, "PDDL2.1: an extension to PDDL for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.

[36] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager, "SWI-prolog," *Theory and Practice of Logic Programming*, vol. 12, no. 1-2, pp. 67–96, 2012.