Systems and Computers in Japan, Vol. 33, No. 11, 2002
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J84-D-II, No. 8, August 2001, pp. 1741–1752

Recognition of Indoor Images Employing Supporting Relation between Objects

Keiji Yanai[1] and Koichiro Deguchi[2]

[1]Department of Computer Science, The University of Electro-Communications, Tokyo, 182-8585 Japan

[2]Graduate School of Information Sciences, Tohoku University, Sendai, 980-8579 Japan

SUMMARY

In this paper, we describe a new design of a recognition system for a single image of an indoor scene including complex occlusions. In conventional works, the systems could not recognize images of an indoor scene including complex occlusions. Our system can treat them by employing supporting relation between objects. In our system, first, the system estimates the 3D structure of an object by fitting a 3D structure model to the image qualitatively. Next, by checking the supporting relation between objects, it eliminates object candidates that cannot exist and estimates real objects from their parts in the image. Finally, the system recognizes objects that are compatible with each other. We implemented the system as a multi-agent-based image understanding system. In this paper, we describe the design of the system and results of experiments. © 2002 Wiley Periodicals, Inc. Syst Comp Jpn, 33(11): 14–26, 2002; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.10142

Key words: ordinary object recognition; scene understanding; image understanding system; supporting relation.

Contract grant sponsor: Supported in part by a grant from the Inamori Foundation.

1. Introduction

The objective of our research is to realize a system that recognizes objects and their spatial relations in a single image of a real-world scene including complex occlusions. In our research, "object recognition" means to obtain the category name of the object, such as "desk" and "chair," from an image of a real-world scene. In this paper, we restrict the recognition target to an indoor scene, and we propose a new recognition method for treating complex occlusions.

In a usual indoor scene, various objects are piled up. For example, there exists a desk on the floor and a book on the desk (Fig. 1). Therefore, many occlusions occur, and the

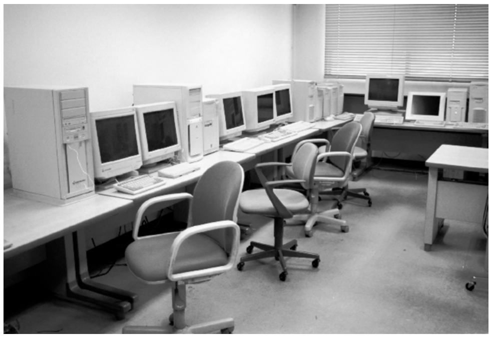


Fig. 1. An example of an image of an indoor scene.

recognition of an indoor image must deal with them. In many conventional investigations of the recognition of a scene including occlusions, an exact shape model of a target object is used to recognize a single object. They fit the model to partial features and estimate the total appearance of the target object. In such a way, these approaches are effective for recognition of objects that have constant shapes such as industrial parts. For real-world scenes, however, it is impossible to prepare exact 3D models of the target objects in advance, since objects represented by one category name have many different 3D shapes. Therefore, we use structural models of objects for the recognition of real-world scenes. For such a scene including complex occlusions as in Fig. 1, however, it is difficult to recognize all objects even with structural models.

In our research, using effectively the spatial relations between objects that are not exploited in conventional recognition systems for a single object, we propose a new recognition method for a scene including complex occlusions when exact shape models are not obtained in advance.

Our previously proposed system [14, 15] and conventional image understanding systems for a single real-world image including multiple objects, such as the Schema System [4], make use of spatial relations among objects as a recognition clue. Since their targets are landscape or aerial images, they perform recognition by region segmentation and labeling. Basically, they treat a target image as a 2D scene and interpret 2D spatial relations on the image even for a 3D scene. Such 2D analysis for a 3D scene is effective in aerial images and distant view images of an outdoor scene, since in such scenes the depth of target objects is much smaller than the distance to the scene. However, for short-range view images such as an indoor image, 2D analysis is not effective, because by 2D analysis the system cannot determine front-and-back and top-and-bottom relations between objects. The depths of target objects in a short-range view scene cannot be ignored. Therefore, 2D analysis cannot make the best use of spatial information among objects, and is not effective in handling complex occlusions.

On the other hand, since humans have knowledge of the rough 3D structure of objects existing in the real world, even from a single image they can estimate the 3D structure of objects and 3D spatial relation among them. A human also has a knowledge of physical laws, such that an object without support falls. Thus, for example, if a personal computer is on a flat board whose legs are not seen, we can infer from our knowledge that the flat board has legs and is a part of a desk, that there is a floor under it, and that the floor supports the desk. For a system that understands a 3D scene from a single 2D image, it is necessary to provide a 3D spatial reasoning ability based on qualitative physical laws.

In our research, for the recognition of scenes including complex occlusions, we adopt 3D model fitting and reasoning about the "supporting relation," which is different from simple 2D recognition by region segmentation and labeling. First, we estimate the 3D structure of objects by qualitative fitting of 3D structure models that reflect the functionality of the objects themselves. Then, we reason about the "supporting relation" between objects, which is the relation in which an object is on another object.

Model fitting is first carried out for such image features as edges and regions extracted from an image. In this research, since our main interest is an ability to deal with real-world objects that have various shapes and views, we adopt a method by which the variety of target object views can be handled. Then we eliminate object candidates that cannot exist and estimate actual objects from their parts seen in the image by inferring supporting relations, which enables us to compensate the uncertainty of estimation of the 3D structure of objects. In this way, we can recognize objects in an image of an indoor scene including complex occlusions without exact shape models.

In this paper, we describe our approach to introducing qualitative 3D model fitting and the "supporting relation" checking mechanism, and next describe the knowledge of relations between objects, a method of evaluation of object candidates based on it, and an implementation based on our multiagent-based image understanding system, which is called MORE (Multiagent architecture for Object REcognition) [14, 15]. Finally, we present the results of recognition experiments.

2. Recognition of an Individual Object

Since objects represented by such nouns as "desk" and "chair" have various shapes, it is impossible to prepare exact shape models in advance. Therefore, in this research, to recognize each single object, we prepare prototype models that represent essential functional structures common to the same kind of objects [5, 10, 11]. For example, the functional structure of a "chair" is a combination of a sitting surface and one or four legs, and that of a "desk" is a combination of a desk face and four legs. By fitting models to image features extracted from an image and estimated from "supporting relations," we infer the existence of objects and the 3D structure of object candidates. For some objects we prepare several prototype models, and we select the model with the highest confidence value from among them.

Although such a simple model fitting method enables us to recognize objects with various shapes, it sometimes confuses different kinds of objects and creates conflicts. In this case, we resolve the conflict by evaluation of relations to other objects as well as accuracy of model fitting, and identify the target objects.

2.1. Representation of a model

A prototype model is represented by **model elements** and a **model graph**. Model elements are polygons and straight line segments according to the appearance of the object [Fig. 2(a)], which carry information about their real shape and their generally expected pose in the real world. A model graph represents connecting relations between the model elements [Fig. 2(b)]. In Fig. 2(b), the model elements of the desk are defined as a parallelogram (PG) with four vertices (fl, fr, rr, rl) and a vertical line (VL) with two vertices (t, b). The lengths of two edges of the parallelogram are *a* and *b*, respectively, and the length of the vertical line is *c*. These two model elements are defined as a horizontal rectangle plane and a vertical bar in the real world. The model graph in Fig. 2(b) indicates that an upper vertex(t) of each vertical line is connected to each vertex of a parallelogram. In such a way, we define a "desk" model as consisting of a horizontal plane and vertical legs [Fig. 2(e)], and a "chair" as consisting of a horizontal sitting plane and vertical legs, so that the common structures of many "desks" and "chairs" are well represented by these models.

Each model has information on a range of relative sizes of model elements and properties regarding which elements are supportable and which are to-be-supported [Fig. 2(c)]. Here, a "**supportable**" **element** and a "**to-be-supported**" **element** mean an element that can support other objects and an element that must be supported by another object, respectively. These properties are used in the stage of checking supporting relations described later. For example, the model of "desk" has one parallelogram (PG) as its supportable element and four vertical line (VL) segments whose bottoms are its to-be-supported elements as shown in Fig. 2(f). Figure 2(d) shows the weight value of each model element used evaluating the confidence value of an object candidate.

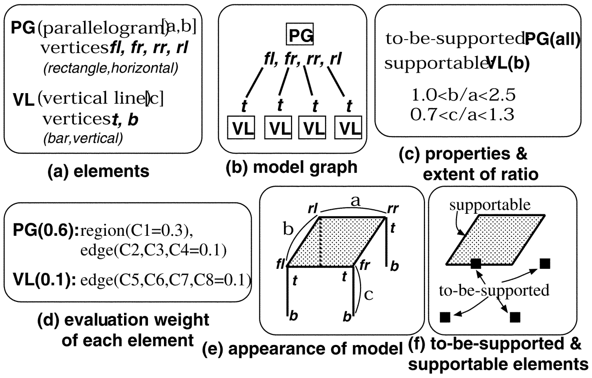


Fig. 2. An example of model representation of "desk."

2.2. Method of fitting a model to the image

For detecting an object candidate, we first extract line segments and regions from the image. These line segments and regions are called **basis elements**, since they are evidence for fitting a model. We extract them by conventional image processing algorithms, for example, the Canny edge detector [2], the Hough transformation, the region growing segmentation method, and so on. Next, we group line segments and regions into proximity pairs of lines, parallel pairs of lines, U-shapes, which consist of a parallel pair of lines and their connecting line, and parallelograms by the perceptual grouping method [7].

We infer the qualitative 3D structure as well as the existence of an object candidate by fitting a model to groups of line segments and regions extracted from the image (Fig. 3). We judge whether a model is applicable or not by examining whether the range of sizes of all elements assigned to a model beforehand is satisfied and whether the **confidence value of a candidate**, which represents goodness of fit, exceeds a threshold value. If fitting is successful, an **object candidate** is generated.

In addition, we estimate which parts are vertical, horizontal, supportable, and to-be-supported by using the properties of model elements related to the categories of vertical, horizontal, supportable, and to-be-supported. Although this model fitting is qualitative without quantitative accuracy, we can obtain the size and direction of an object candidate accurately enough to carry out qualitative reasoning about supporting relations. We call the regions and line segments of an object candidate estimated by fitting a model "**candidate elements**." Candidate elements are all elements expected to be viewed without occlusions.

The **confidence value of a candidate** used in resolving conflicts among models is scored from 0 to 1. It is

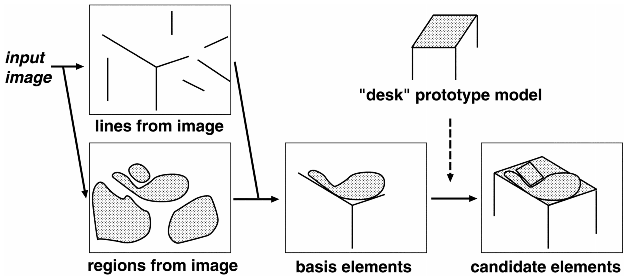


Fig. 3. Flow of estimating an object candidate of "desk."

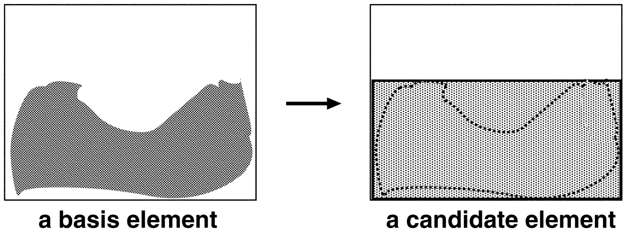


Fig. 4. Estimating an object candidate of "floor."

defined as the weighted sum of the ratios of the corresponding basis elements to the candidate elements, as follows:

$$V_{im} = \min\left(\sum_{i=1}^{n} C_i \frac{b_i}{e_i}, 1\right) \tag{1}$$

where n is the number of elements, b_i is the number of pixels of the i-th basis element, and e_i is the number of pixels of the i-th candidate element. C_i is a weighting factor, which represents the degree of importance of each element and is provided as a priori information with each model. For example, "desk" has six elements, a contour and a region of a desk face and each leg, and the W_i for them are set as 0.4, 0.3, and 0.1, respectively.

We fit a plane model to background objects, such as floors, walls, roads, and sky, that have no constant shape, as shown in Fig. 4. For these background objects, the model elements are only plane regions. We compute confidence value of a candidate for them as follows:

$$V_{im} = b/e \tag{2}$$

where b is the number of pixels of the region of a basis element, and e is the number of pixels of the region of a candidate element.

3. Checking Supporting Relation

The "**supporting relation**" is the relation that an object is supported by other objects. All objects except background objects, such as floors, walls, roads, and sky, must be supported by other objects in the real world due to gravity. Humans know such qualitative physical laws empirically, and they may have an effect on human perception. Then, according to this fundamental rule of real-world scenes, the system makes use of relations regarding which object supports which object.

Each time the system generates a new object candidate, it examines whether the "supporting relation" holds between already generated candidates and the new one. By checking the supporting relation between objects, the system eliminates object candidates that cannot exist and estimates actual objects from their parts seen in the image.

The "supporting relation" holds when an object can be considered to be located on another object and to be supported by it. Checking of the "supporting relation" is carried out by examining whether the regions of to-be-supported elements of the object are almost all included in the region of supportable elements of another object. If so, the former object is regarded as supported by the latter one. We present an example including the relations that "`desk is supported by floor`" and "`book is supported by desk`" in Fig. 5.

In Fig. 5, we can also consider the book to be supported directly by the floor. However, we can infer *floor* ⇒ *desk* ⇒ *book*, from *floor* ⇒ *desk*, *floor* ⇒ *book*, and *desk* ⇒ *book*, where the relation that object A supports object B is defined as $A \Rightarrow B$.

If a candidate has no supporting relation, to-be-supported elements of the candidate are regarded as "**virtual basis elements**," and the system searches for a new candidate with supportable elements including the virtual basis elements (Fig. 6). This search is carried out for kinds of objects that have the possibility of supporting the candidate

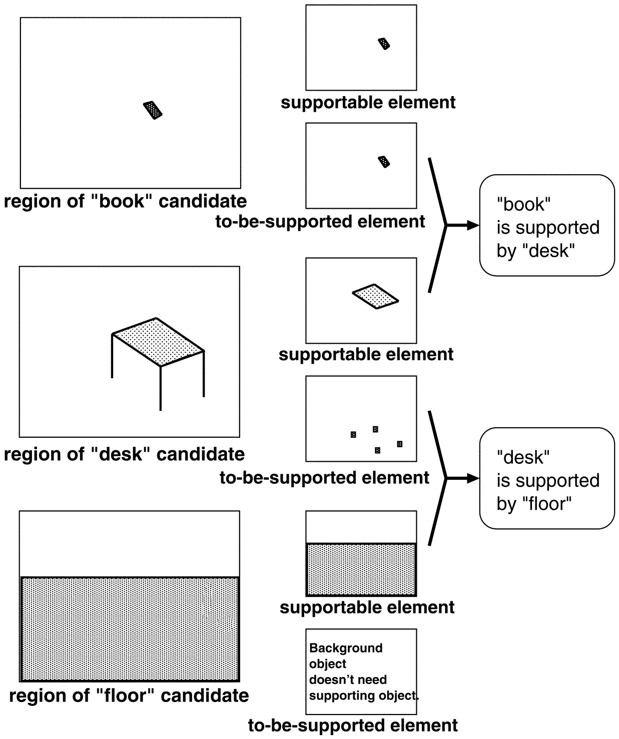

supportable element
region of "book" candidate
to-be-supported element
"book" is supported by "desk"
supportable element
region of "desk" candidate
to-be-supported element
"desk" is supported by "floor"
supportable element
Background object doesn't need supporting object.
region of "floor" candidate
to-be-supported element

Fig. 5. Checking "supporting relation."

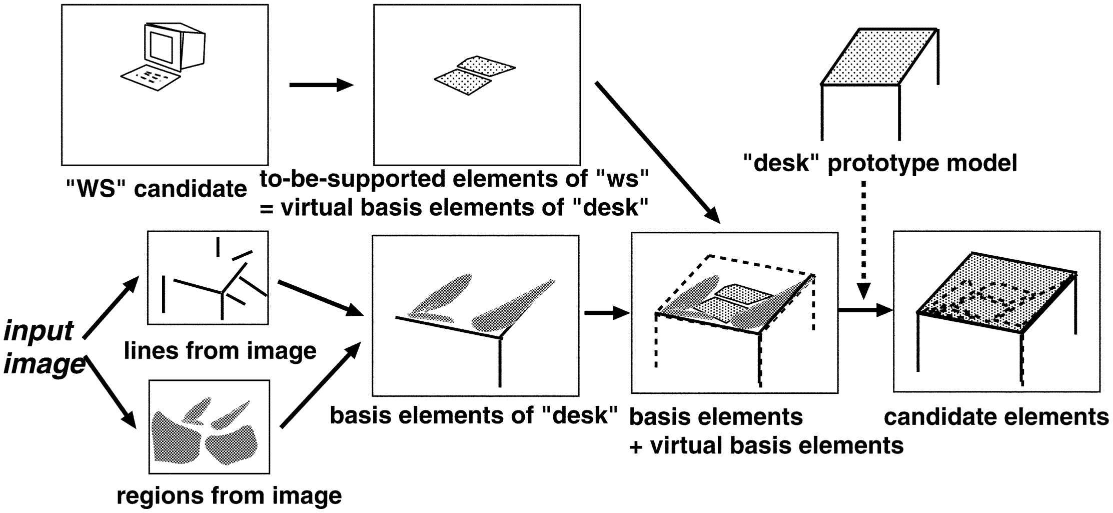


Fig. 6. Estimating a "desk" candidate that supports the "workstation" candidate by combining virtual basis elements and basis elements.

with no supporting relation according to the "relational knowledge" that we shall describe later. Thus, the system can detect a new candidate that could not be detected before.

For example, if the system generates a "workstation (WS)" candidate with no supporting relation, it regards the to-be-supported elements of "WS" as the virtual basis elements of a desk face element (Fig. 6). The system searches the surrounding area of the virtual basis element for a new candidate including it as a supportable element, and, as a result, detects "desk" by fitting a model to the basis elements extracted from an image and the virtual basis elements generated from the to-be-supported elements of "WS."

In this way, by introducing the "supporting relation," the system recognizes an object occluded by another object.

If candidates other than background objects have no supporting relation, finally, the candidates are canceled.

4. Evaluation of Relational Knowledge and Object Candidates

4.1. Relational knowledge

The system processes "relational knowledge" beforehand. In our system, it consists of descriptions of the relative spatial relations generally expected between two objects. It is used for computing the confidence value of relations and inferring a new object candidate that has supporting relation to already detected candidates.

Relational knowledge is represented by a combination of "name of relation," "source object's name," and "destination object's name." For example, "`on(book, desk)`" and "`next-to(chair, desk)`" mean "a book is usually on a desk" and "a chair is usually next to a desk," respectively. At present, we have two types of relations, "on" and "`next-to`."

The system judges whether each relation holds by the information on supporting relations and the relative location of objects in the image. This judgment is made by checking the relational knowledge one by one. "`on(A, B)`" holds when object A is supported by object B. "`next-to(A, B)`" holds when both objects A and B are supported by the same one object, and are located on the same support-capable plane. The system uses all relational knowledge to ascertain whether relations are applicable.

The **confidence value of relations** V_{re} is determined as a weighted sum of holding relations:

$$V_{re} = 1 - \exp\left(-k\sum_{i=1}^{r} c_i n_i\right) \qquad (3)$$

where r is the total number of relations, n_i represents whether relation i holds, taking a value of 0 or 1, and k is a constant. We set k to 0.4 for the experiments. c_i is a weighting factor that represents the significance of each relation. At present, it is set 1.0 for the relation "on" and 0.5 for the relation "`next-to`."

Equation (3) includes an exponential term, so that the increase of V_{re} is much larger when the number of holding relations increases from one to two, compared to the case in which it increases from five to six. The confidence value of relations V_{re} takes a value from 0 to 1. It is estimated by checking relational knowledge items one by one. It can be regarded as the degree of naturalness of existence of a candidate in the scene.

4.2. Evaluation of object candidates

If two or more agents generate different candidates in the same region of an image, a conflict occurs. Conflicting candidates are compared by confidence value V. V is calculated as a weighted sum of the confidence value of a candidate V_{im} and the confidence value of relations V_{re} as follows:

$$V = (V_{im} \times S' + V_{re} \times w)/(S' + w) \qquad (4)$$

$$S' = \min(S, 2w) \qquad (5)$$

where S is the total number of pixels of candidate elements and w is a constant that controls the balance between V_{im} and V_{re}. The weights are tuned by taking the size of a candidate into account. If the number of pixels S of the candidate is more than $2w$, the ratio of the weight of V_{im} and V_{re} is 2 : 1. Otherwise, it is $S : w$. This emphasizes V_{re} of the candidates having a small number of pixels when evaluating V. We set w to 2500 for the experiments.

As a result of comparison of V, the candidate with the highest value remains, and all other candidates are eliminated.

There are no established confidence values of object candidates in an image understanding system, and various evaluation methods have been used in many systems. We modified the method used in Ref. 5 and applied it for our system.

5. Overview of the System

5.1. System architecture

We designed the system based on "MORE" (Multiagent architecture for Object REcognition), which we proposed in Refs. 14 and 15. "MORE" is a multiagent-based architecture. The system is constructed as an assembly of agents that recognize a single kind of object from an image separately, and it has no central management mechanism (Fig. 7). This architecture enables the system to recognize various different kinds of objects by adding agents. Since we can apply different knowledge representation and recognition methods for each agent, it is suitable for implementing a large-scale image understanding system. In addition, a nonfixed processing flow allows flexible fusion of top-down and bottom-up processing.

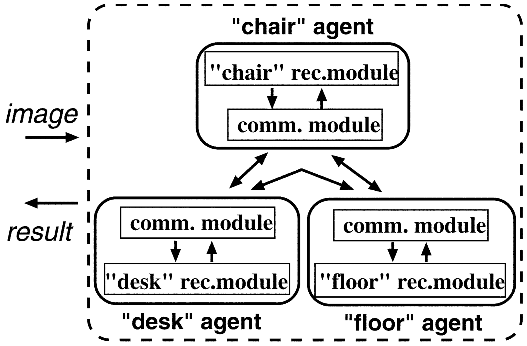


Fig. 7. Basic structure of the system.

One agent consists of a recognition module which recognizes a single kind of object, and a communication module which cooperates with other agents.

A **recognition module (RM)** has models of objects, and recognizes only one kind of target object in an input image. Each time an RM finds a new object candidate, it computes its confidence value of a candidate, then sends information about the basis elements, candidate elements, supportable elements, to-be-supported elements, and the confidence value of a candidate to the communication module in the same agent. In our architecture, we do not assume a special implementation for an RM. A suitable recognition module is implemented by using suitable recognition methods, algorithms, knowledge, and knowledge representation.

A **communication module (CM)** carries out negotiation among the agents so as to maintain consistency over the whole system. A CM receives information about object candidates from the RM, exchanges it with other CMs, and resolves conflicts by comparing confidence values. Every CM has relational knowledge. Using this knowledge, it computes the confidence value of relations, and it checks supporting relations with respect to candidates generated by other agents.

5.2. Processing flow

The processing flow among all of the modules is message-driven. We describe the detailed flow of messages for the example of Fig. 8.

(1) An input image is sent to the recognition modules of all of the agents. Each CM sends an "**initial recognition request**" to each respective RM. Then, each RM starts the recognition.

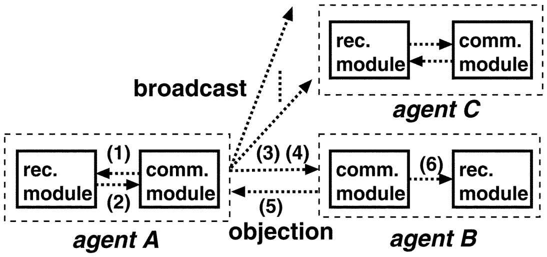


Fig. 8. Flow of messages. (1) Initial recognition request. (2) Information on a new object candidate. (3) Broadcasting information of a new object candidate. (4) To-be-supported request message. (5) Objection message. (6) Re-recognition request.

(2) Each time an RM finds an object candidate, it sends information on the candidate and its candidate confidence value to the CM.

(3) The CM checks whether the candidate is supported by any other candidates, and broadcasts the information on the candidate to all other agents.

(4) If a candidate other than a background object is not supported by any object candidate, the CMs generate virtual basis elements and broadcast their information as a "**to-be-supported request**."

(5) Other agents examine whether the broadcast is consistent with their own object candidates. If not, the agents sends back an "**objection message**." Then conflict resolution is performed between the CMs concerned.

(6) If a CM receives a "to-be-supported request" and has relational knowledge that the candidate is usually on its own object, the CM sends a "**conditional recognition request**" to its RM. The RM starts to find object candidates with supportable elements overlapping the virtual basis elements of the candidate.

In addition, our system has a "revival mechanism." The system revives an object candidate that was eliminated once, when its confidence value of relations is changed and the result of conflict resolution becomes invalid. By this mechanism the system always keeps object candidates that are consistent with others, and maintains consistency among agents. If all modules of all of the agents are in the state of waiting for a message and there is no message on the communication lines, the whole recognition activity of the system terminates.

6. Experiments

We implemented an experimental system with six agents ["desk," "chair," "wall," "floor," "book," and "workstation (WS)"] on a PC cluster system consisting of six PCs (Intel Celeron 450MHz with 128 MB memory) using the PVM library [6]. In this experimental system, since we have not yet taken account of effective load balancing, each agent is implemented on each PC, and all recognition modules extract basis elements from an input image independently. It is possible to share processing of basis element extraction. In this section, we give a description of the processing of two sample images of a relatively simple indoor scene and a more complex scene, and describe the experimental results for 20 indoor images.

6.1. Example of recognition

After the system is given an input image, which is a 256-level gray-scale image, the image is sent to the RMs in all agents, and the RMs start extracting straight line segments and regions as image features. When sample image No. 1 (Fig. 9, 480 × 360) is given, first, straight line segments (Fig. 10) and regions (Fig. 11) are obtained by extraction processing of image features such as edge detection, line detection, and region segmentation. Next, the system extracts proximity pairs of lines, parallel pairs of lines, U-shapes, which consist of a parallel pair of lines and their connecting line, and parallelograms by grouping straight line segments (Fig. 12). The RM of the "WS" agent extracts two parallelograms as front parts of display monitors of workstations as the most prominent features and one parallelogram and one U-shape for parts of keyboards. The RM of the "WS" agent fits a model of the front part of the display and a model of a keyboard to the parallelograms with two vertical edges and the parallelograms in the front of a display, respectively, and generates two new object candidates (Fig. 13). The right-side parts of WS candidates are detected from vertical straight line segments behind the front parts of the display monitors. In addition, the RM infers parallelogram elements from the bottom straight edges of the front sides and right sides of displays, and considers the region consisting of them and the region of the keyboards as to-be-supported elements of the "WS" candidates (Fig. 14). Thus, the "WS" agent generates two

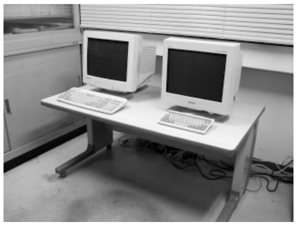


Fig. 9. Sample image No. 1.

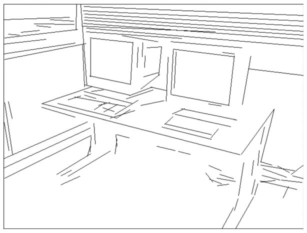


Fig. 10. Extracted straight edges.

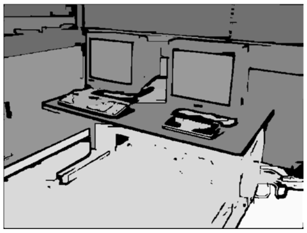


Fig. 11. Extracted regions.

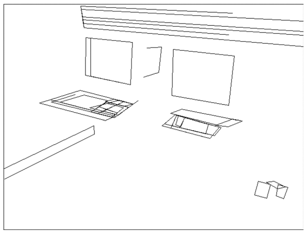


Fig. 12. Parallelograms and U-shapes.

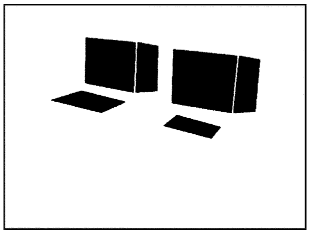


Fig. 13. Two "WS" candidates.

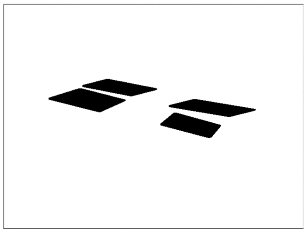


Fig. 14. To-be-supported elements of two "WS" candidates.

new candidates, but neither of them is supported by any object candidates. Then, the CM of the "WS" agent broadcasts not only the information on the two new candidates, but also "to-be-supported request" to other agents.

At first, the "desk" agent detects no desk candidates, since the obtained image features are only straight edges and regions, which constitute insufficient information (Fig. 15). However, after receiving a "to-be-supported request" with information on virtual basis elements from the "WS" agent, the "desk" agent tries to detect a new candidate that has the virtual basis elements as its basis elements, since it has relational knowledge "`on(desk, WS)`." The "desk" agent detects enough basis elements to generate a new desk candidate by joining the virtual basis elements to some straight edges and regions (Fig. 16). By fitting a model to these basis elements, a desk candidate can be detected (Fig. 17). In addition, a floor is detected correctly by a "to-be-supported request" from the "desk" agent. Finally, two "WSs," two "desks," and a "floor" are detected as shown in Fig. 18.

We next describe an experiment with sample image No. 2 (Fig. 19, 640 × 480) which is a more complex scene than sample image No. 1. In this experiment, first, four WSs are detected (Fig. 20). Their to-be-supported elements are shown in Fig. 21. At the same time, "books" are detected, and 10 "book" candidates are generated (Fig. 22). Then, three WSs, excluding the second WS from the right generate a conflict with books. By conflict resolution, three WSs remain and books are eliminated. For example, in the conflict between the keyboard of the rightmost WS and the book, the candidate confidence values V_{im} are 0.78 and 0.59, the confidence values of relations V_{re} are 0.33 and 0.33, and total confidence values V are 0.62 and 0.43, respectively. As a result, the book candidates are eliminated. The real keyboard of the second WS candidate from the right is recognized as a book candidate, since the position of its keyboard part is detected incorrectly.

Since the WS candidates and the remaining book candidates have no supported object candidates, "to-be-supported requests" are issued. Then, the "desk" agent joins virtual basis elements that originate from the to-be-sup-

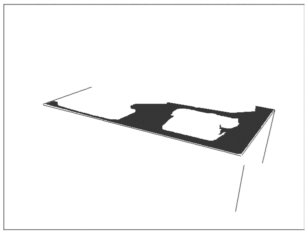


Fig. 15. Basis elements of a "desk."

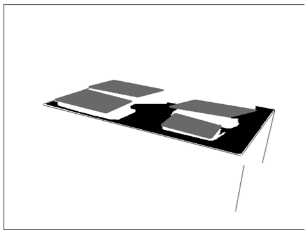


Fig. 16. Virtual elements and basis elements of a “desk” candidate.

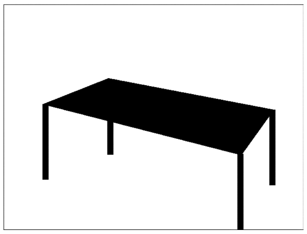


Fig. 17. A “desk” candidate.

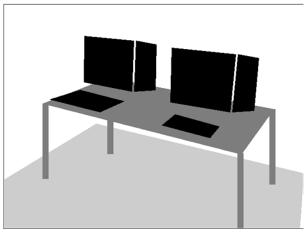


Fig. 18. Recognition result.

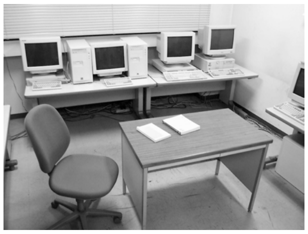


Fig. 19. Sample image No. 2.

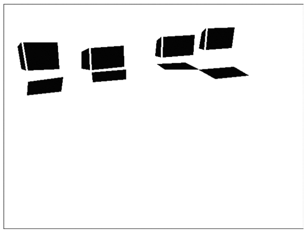


Fig. 20. Four “WS” candidates.

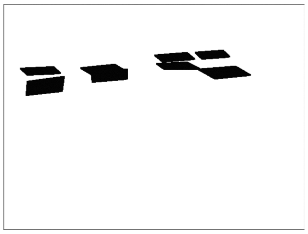


Fig. 21. To-be-supported elements of four “WS” candidates.

ported elements of WS candidates (Fig. 21) and book candidates to the basis elements of desk candidates (Fig. 23), so that the agent can recognize the desk candidate, although much of the desk face is occluded by other objects (Figs. 24, 25). Three of five book candidates causing no conflict fail to have identified object candidates supporting them, and are eliminated due to no supporting relation. Actually, these book candidates are incorrectly detected, and are eliminated correctly by checking the supporting relation. Finally, “WSs,” “desks,” “books,” a “chair,” and a “floor” are recognized correctly, although two “books” on the right-back desk are recognized incorrectly as shown in Fig. 26. In this experiment, we implemented “WS” agent to recognize its candidate as a pair consisting of a display and a keyboard, so that two main bodies on the right-back desk are not recognized.

6.2. Experimental results

We performed recognition experiments on 20 indoor images whose size was 480 × 360. We show 12 out of the 20 images used in this experiment in Fig. 27. In this figure, the images in the upper row are quite simple, those in the lower row are complex, and those in the middle row have intermediate complexity. The numbers of images in each category are 7, 7, and 6, respectively. We evaluate the results by dividing them into three grades, “almost entirely cor-

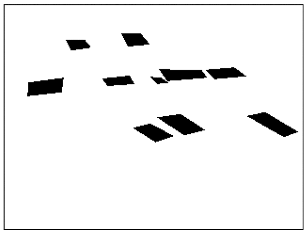


Fig. 22. “Book” candidates.

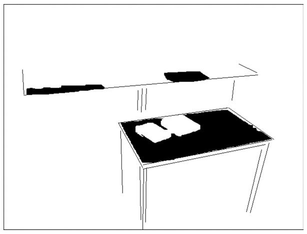


Fig. 23. Basis elements of “desk” candidates.

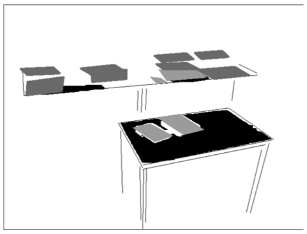


Fig. 24. Virtual basis elements and basis elements of “desk” candidates.

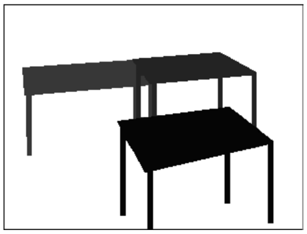


Fig. 25. Three “desk” candidates.

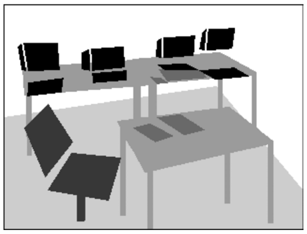


Fig. 26. Recognition result.

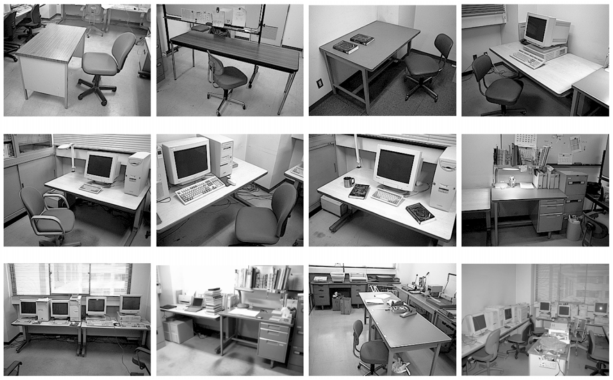


Fig. 27. 12 out of 20 images for experiments. Images in the upper row are quite simple, those in the lower row are complex, and those in the middle row have intermediate complexity.

rect,” “half correct,” and “almost entirely incorrect.” The numbers of images in these three grades are 9, 6 and 5, respectively, as shown in Table 1.

6.3. Discussion

In the experiment with 20 indoor images, an example of “almost entirely correct” images is sample image No. 1 shown in Fig. 9. A “desk” is recognized by checking the supporting relations and recognition with virtual basis elements, although most of the desk face is covered by two “WSs.”

Figure 28 is an example of “half correct” images. In this image, there are a “notebook PC,” piled “books,” an opened “notebook” on the desk, a “bookshelf,” and “book” in the bookshelf. A “notebook PC” and an opened “notebook” are recognized as a “WS” and a “book,” respectively, and a “desk” is recognized correctly by checking supporting relation as shown in Fig. 29. However, since the RM of the “book” agent is implemented to recognize a parallelogram as a book, a parallelogram or a U-shape is not extracted, and the piled “books” and the “book” in the bookshelf are not recognized. If we use the relational knowledge “`on(book, book)`,” if a book on the top is

Table 1. Results for 20 images

almost entirely correct	half correct	almost entirely incorrect
9	6	5

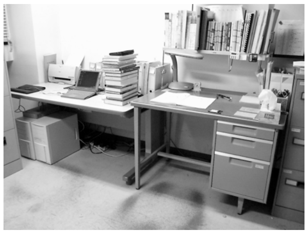


Fig. 28. An example image whose results are half correct.

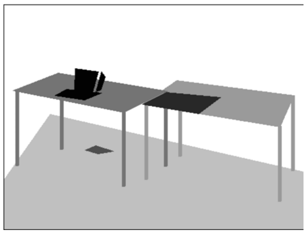


Fig. 29. Recognition result.

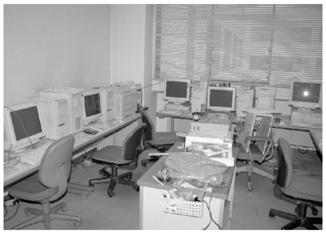


Fig. 30. An example image whose results are almost entirely incorrect.

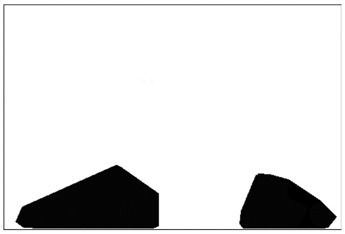


Fig. 31. Recognition result.

not recognized, books under it are also not recognized. To deal with such a case, it is desirable to prepare a model of a block of books in contact with each other in advance.

An example of "almost entirely incorrect" images is shown in Fig. 30, and its result is shown in Fig. 31. This image is so complex that only parts of "floor" are recognized. Although the display of a "WS" appears clearly in the image, the color of the keyboard on the desk is so similar to the color of the desk face that the system cannot recognize the "WS." This depends on the resolution of the input images, so that it is desirable to introduce multiresolution analysis, which can recognize the rough structure of the scene for a lower-resolution image first, and will later recognize details for only the needed parts of a higher-resolution image. There exist unknown objects on the center desk, for which models are not prepared. A wrapper without constant shape is on the front side of the center desk, and it is difficult to represent an object without constant shape such as a wrapper for rigid model fitting. It is necessary to introduce active contour and textural analysis methods.

7. Related Works

The objective of our work is scene recognition when exact models of target objects are not available in advance. The work conducted by Tenenbaum and Barrow [13], in which segmented regions were labeled by the relaxation method, had similar objectives to our work. But their work used too simple methods, and it was not usable for complex images. After that, knowledge-based recognition systems, for example, that of Ohta [9], the Schema System [4], and SIGMA [8], appeared. They used both models for single objects and relational knowledge among objects, and achieved an integration of bottom-up and top-down processing. Our work is also similar to theirs, but their target was not indoor images but outdoor images or aerial images that seldom include occlusions and need not be treated in three-dimensional space, because their method of utilizing spatial information is two-dimensional.

The features of our system are that we estimate supportable and to-be-supported elements by fitting 3D structure models to partial features extracted from an image, and verify the generated object candidates by checking supporting relations. A similar system, which generates individual object candidates and verifies them based on relations among candidates, is CONDOR [12]. Its method of recognition is called context-based recognition. However, CONDOR employs two-dimensional recognition for images of an outdoor scene. It does not employ qualitative three-dimensional recognition like our system.

Cooper and colleagues [3] and Brand [1] realized recognition with consideration of the physical relation that an object supported another object against gravity. In their work, the systems analyzed scenes based on knowledge of qualitative physical laws. However, the objective of their work was not recognition but understanding of physical action from images. Therefore, the target objects were simple objects like blocks.

In our system, in recognizing an individual object, such as a "desk" or "chair," represented by a general noun, we pay attention to the essential structure that provides the functions of an object. This idea is imported from function-based recognition [10, 11]. It claims that the essence of an artifact is to provide functions to humans and that we should recognize an artifact by examining whether it has a shape providing a relevant function.

8. Conclusion

In this paper, we have proposed a system that estimates the 3D structure of a target object by fitting a qualitative model qualitatively, and verifies object candidates by checking the "supporting relation" using "supportable regions" and "to-be-supported regions." It totally realizes flexible recognition for real-world indoor images including complex occlusions. We have implemented the system as a multiagent-based image understanding system on a PC cluster.

In the present system, we often obtain imperfect results due to insufficient ability of the recognition modules. Thus, in future work, we plan to study how to construct a recognition module with greater ability and more effective cooperation mechanisms and learning mechanisms to make up for the insufficient ability of individual recognition modules. We also plan to make use of high-resolution images recently obtainable by digital cameras easily in order to recognize objects that appear with small size in an image.

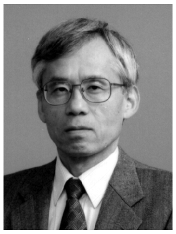
Experiments using the current implementation revealed no problems of convergence of recognition processing and stability of recognition results. In systems of larger scale by increasing the number of agents in the future, however, we will need to analyze behavior of the system in this respect.

Acknowledgment. Part of this work was supported by a grant from the Inamori Foundation.

REFERENCES

1. Brand M. Physics-based visual understanding. Comput Vision Image Understanding 1997;65:192–205.
2. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Machine Intell 1986;5:140–150.
3. Cooper PR, Birnbaum LA, Brand ME. Causal scene understanding. Comput Vision Image Understanding 1995;62:215–231.
4. Draper B, Collins R, Brolio J, Hanson A, Riseman E. The schema system. Int J Comput Vision 1989;3:209–250.
5. Kim D, Nevatia R. Recognition and localization of generic objects for indoor navigation using functionality. Image Vision Comput 1998;16:729–743.
6. Geist A, Beguelin A, Dongarra J, et al. PVM: Parallel virtual machine. MIT Press; 1994.
7. Lowe DG. Perceptual organization and visual recognition. Kluwer Academic; 1985.
8. Matsuyama T, Hwang VS. SIGMA: A knowledge-based aerial image understanding system. Plenum Press; 1990.
9. Ohta Y. Knowledge-based interpretation of outdoor natural color scenes. Pitman Advanced Publishing Program; 1985.
10. Stark L, Bowyer K. Achieving generalized object recognition through reasoning about association of function to structure. IEEE Trans Pattern Anal Machine Intell 1991;13:1097–1104.
11. Stark L, Bowyer K. Function-based generic recognition for multiple object categories. Comput Vision Image Understanding 1994;50:1–21.
12. Strat TM, Fischler MA. Context-based vision: Recognizing objects using information from both 2-D and 3-D imagery. IEEE Trans Pattern Anal Machine Intell 1991;13:1050–1065.
13. Tenenbaum JM, Barrow HG. Experiments in interpretation guided segmentation. Artif Intell 1977;8:241–274.
14. Yanai K, Deguchi K. An architecture of object recognition system for various images based on multiagent. Proc 14th Int Conf Pattern Recognition 1998;1:278–281.
15. Yanai K, Deguchi K. A multi-agent architecture of object recognition system for various images. Trans Inf Process Soc Japan 1998;39:170–177. (in Japanese)

AUTHORS (from left to right)



Keiji Yanai received his B.E. and M.E. degrees from the University of Tokyo in 1995 and 1997. He then became a research associate in the Department of Computer Science of the University of Electro-Communications. His recent research interests include an image understanding system and a multimedia database.

Koichiro Deguchi received his B.E., M.E., and D.Eng. degrees from the University of Tokyo in 1974, 1976, and 1980. From 1984 to 1987, he was an associate professor in the Department of Information Engineering, Yamagata University, and from 1988 to 1998, an associate professor in the Department of Mathematical Engineering and Information Physics, University of Tokyo. Since then, he has been a professor in the Graduate School of Information Sciences, Tohoku University. His recent research interests include computer vision, signal processing, and multiprocessor architecture for them.