IEICE TRANS. INF. & SYST., VOL.E84–D, NO.12 DECEMBER 2001

PAPER *Special Issue on Machine Vision Applications*

A Multi-Resolution Image Understanding System Based on Multi-Agent Architecture for High-Resolution Images

Keiji YANAI† ***and*** **Koichiro DEGUCHI**††, ***Regular Members***

SUMMARY Recently a high-resolution image that has more than one million pixels is available easily. However, such an image requires much processing time and memory for an image understanding system. In this paper, we propose an integrated image understanding system of multi-resolution analysis and multi-agent-based architecture for high-resolution images. The system we propose in this paper has capability to treat with a high-resolution image effectively without much extra cost. We implemented an experimental system for images of indoor scenes.
key words: *multi-resolution analysis, multi-agent, image understanding system, scene interpretation*

1. Introduction

Recently a high-resolution image that has more than one million pixels is available easily due to high-performance digital cameras. However, in researches of image understanding, since such a high-resolution image requires much time and memory to process, the image is usually reduced to a lower-resolution image that has ten thousands of pixels. It may throw away significant information included in the high-resolution image. The feasible practical solution is to exploit multi-resolution analysis for high-resolution images, where we use a low-resolution image and recognize rough structure of scene at first, and use only the needed parts of a higher-resolution image later.

Multi-resolution analysis was originated as works of image processing such as edge detection and region segmentation in 1980s, and later it was applied to image recognition systems. In works by Z.Li [1] and by C.L.Tan [2], an image pyramid was introduced by reducing resolution of a high-resolution input image in several steps. Then, first, the rough structures were extracted from low-resolution image, and next significant parts of higher-resolution images were selected and processed based on the rough structure. However, the systems they implemented are very restricted, because of lower ability of computers of those days. Nowadays since computers have made rapid progress, we can realize more large-scale and complicated system.

Here, we introduce multi-resolution analysis to the multi-agent-based image understanding system [3], [4] for effective use of high-resolution images. Our multi-agent-based architecture for an image understanding, MORE (Multi-agent architecture for Object REcognition), is suitable for large-scale and complex recognition system due to its flexible and extensible architecture.

Almost conventional object recognition systems with multi-resolution analysis aimed at recognizing a single object. On the other hand, the objective of our system is to recognize multiple objects in a single image of real-world scene including complex occlusions. In our research, the "recognition" means to obtain a category name of the object, such as "desk" and "chair," from an image of real-world scene.

Most of the existing object recognition systems with multi-agent architecture, for example, the Schema System [5] and SIMGA [6], didn't adopt multi-resolution analysis. Therefore, target images of those systems were restricted to relatively small size images, and they could not zoom in on significant parts of a target image.

In this paper, we describe design and implementation of a multi-agent-based system employing multi-resolution analysis. By resolution-selection mechanism, the system recognizes objects that couldn't be recognized in a low-resolution image without much extra time and memory. We implemented an experimental system for indoor images on PC cluster system. We also describe experimental results on the system.

2. Introducing Multi-Resolution Analysis

Generally in image recognition systems unless more than a certain quantity of image features such as line and regions are detected in the initial stage, it is impossible to generate object candidates. Then, our and almost other image understanding systems have mechanism of "re-recognition." It means that the system tries again to recognize objects that were not found in the initial stage. However, there have not been established methods of "re-recognition." Many systems only change parameters or thresholds in low-level image processing algorithms.

Manuscript received February 20, 2001.
Manuscript revised June 20, 2001.
†The author is with the Department of Computer Science, The University of Electro-Communications, Chofu-shi, 182–8585 Japan.
††The author is with the Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan.

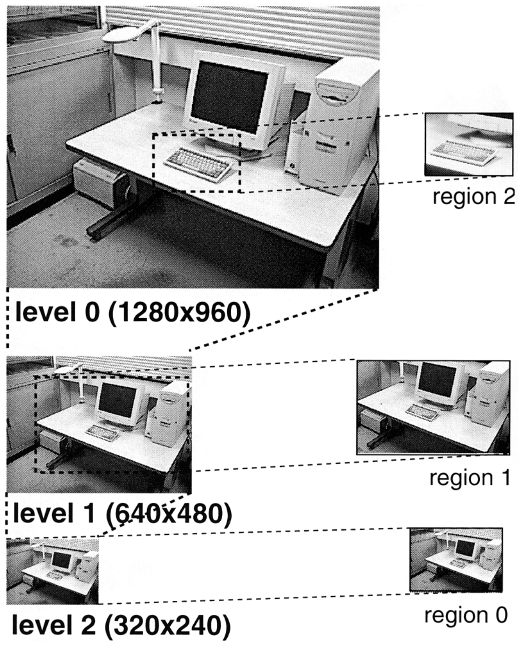


Fig. 1 An image pyramid for multi-resolution analysis.

In case of on-line image capturing, the system can zoom in on needed parts of scene dynamically by methods of active vision [7]. However, it is impossible for the off-line recognition system whose targets is a single still image. Then, we prepare a high-resolution image in advance, at first use the reduced image, and next use some parts of the high-resolution image according to necessity.

Of course, there is an idea that we exploit a whole high-resolution image in the initial recognition stage. Usually, computational costs of image processing algorithms for an image recognition system are higher than $O(n^2)$, where n is the number of pixels in a target image. Therefore, to reduce total data volume by multi-resolution analysis is essential for exploiting high-resolution images.

In our system, we use a lower-resolution image in the initial stage, and in the re-recognition stage we select proper resolution from the image pyramid and process only needed part of an image. An image pyramid is constructed by reducing resolution of an image in several steps by a constant magnification (usually from 0.5 to 0.75) (Fig. 1). We call images in the pyramid an image of level 0, 1, 2, ..., respectively, in order from the original image.

In the re-recognition stage, the system estimates regions where undetected object candidates are expected to exist using already detected candidates and "relational knowledge." It selects resolution of an image in the range where the number of pixels of the region does not exceed a certain threshold value. In short, if a region of interest is determined, resolution level of the image we use for recognition is determined.

Table 1 The number of pixels of each region in each level.

region no.	level 0	level 1	level 2
0	1,228,800	307,200	**76,800**
1	616,224	**154,056**	38,514
2	**54,528**	13,632	3,408

We call this resolution level selected for a certain region "**proper level**" for the region. Proper level l_p is defined as follows:

$$l_p = \arg\max_l \{S_i^l | S_i^l < th\} \tag{1}$$

where S_i^l is the number of pixels of region i in the level l image, and th is a threshold value.

In Fig. 1, the image pyramid consists of three images the resolutions of which are 1280×960, 640×480 and 320×240, respectively. At first, the system analyzes the whole image of the highest level, that is, level 2 (region no.0), and extracts a floor, a desk and a display on the desk. Next, it analyzes the region of the desk candidate extracted from level 0 and its peripheral region in the image of level 1 (region no.1) to examine if there are some objects on the desk. If the region where an object is expected to exist is detected, it analyzes the region using the image of level 0 (region no.2). Then, a keyboard in front of the display can be detected. Table 1 shows the number of pixels of each region in each image level in Fig. 1. In case that threshold th is set as 200,000, proper levels of region no.0, no.1, and no.2 are selected as level 2, level 1 and level 0, respectively. This is because maximum pixel numbers for each region less than this th are selected as shown in Table 1, where the selected numbers are shown in boldface.

3. Recognition Strategy

Targets of our system are scene that consists of artifacts, for example, laboratory scene and PC room scene. The system recognizes each single object and its relations to other objects in an image.

3.1 Recognition of Each Single Object

To recognize each single object, we prepare prototype models that represent essential functional structure common within the same kind of objects [8]. 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. The prototype model is represented by some model elements and a model graph. Model elements are polygons and straight line segments corresponding to appearance of parts of an object (Fig. 2 (a)). They have information about their real shape and their generally expected pose in the real world. A model graph represents connection relations between the model elements (Fig. 2 (b)). Each model has the extent of relative size among each

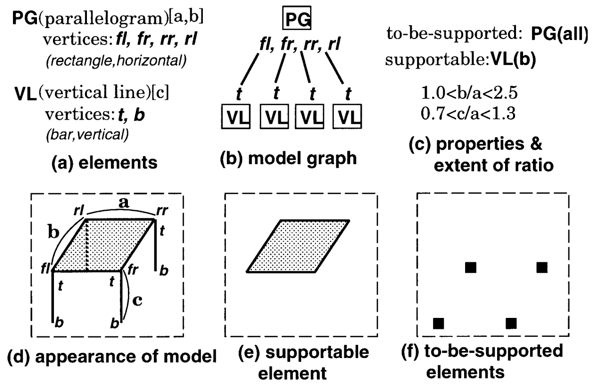


Fig. 2 Model representation of "desk."

elements and information which elements are "supportable" and "to-be-supported" (Fig. 2 (c)). Here, the "supportable" element and the "to-be-supported" element mean that the element can support other objects and the element must be supported by another object, respectively. These properties are used in the stage of "checking supporting relation" described later. For example, the model of "desk" has one parallelogram as its supportable element and four vertical line segments whose bottoms are its to-be-supported elements as shown in Figs. 2 (d), (e), (f).

For estimating regions of an object candidate, first, we extract line segments and regions by conventional methods, for example, Canny edge detector [9], Hough transformation, region growing segmentation method, snake [10] and so on. Next, we search a group of line segments and regions corresponding to each element of a model. We fit the model to the group of line segments and regions extracted from an image (Fig. 3). Here, we call the lines and regions used in model fitting as **"basis edges and regions,"** and the regions where an object is estimated to exist as a **"candidate region."** A candidate region is the total region expected without occlusions. In addition, by using information about "supportable" and "to-be-supported" elements, we estimate **"supportable regions"** and **"to-be-supported regions"** in an image.

We compute confidence value of a candidate V_{im} as a weighted sum of the ratio of basis regions to candidate regions of each element, taking the image level into account:

$$V_{im} = \min\left(\left(\sum_{i=1}^{n} W_i \frac{b_i}{e_i}\right)^k, 1\right) \tag{2}$$

$$k = 1 + l/(l_{max} + 1) \tag{3}$$

where n is the number of elements, b_i is the number of pixels of a basis region or edge, e_i is the number of pixels of a candidate region or edge, l is the level at which the candidate is recognized, and l_{max} is the maximum level of the image pyramid. W_i $(i = 1 \cdots n)$ are weighting factors that represent the degrees of significance of each

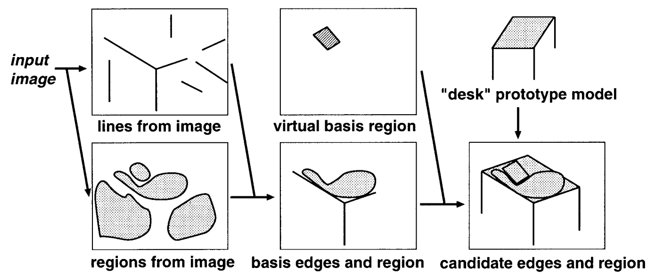


Fig. 3 Estimating a "desk" candidate.

element in each model. Their valves are given as a priori information in each model by hand so that they satisfy $\sum_{i=0}^{n} W_i = 1.0$ within each model.

For example, since "desk" has five elements, desk face and four legs, n is 5, and we set 0.6 for one of W_i and 0.1 for the rest of them, respectively. If the candidate was generated at the levels other than 0, V_{im} becomes k-th power of the weighted sum of the ratio. k is calculated as Eq. (3) and takes the value from 1 to 2. Since V_{im} is less than 1.0, V_{im} becomes the smaller, the higher the level at which the candidate was generated. V_{im} is used for resolving conflict among candidates.

3.2 Checking of Supporting Relation

All objects except background such objects as floor, wall, road, and sky, must be supported by other objects in the real world due to gravity of the earth. We know such physical law empirically, so we can expect existence of a desk under a workstation in the complex scene even if the desk can't be seen. Then, according to this fundamental rule of the real-world scene, the system checks relations which object supports which object. We call such the relation the "**supporting relation**."

Every time the system generates a new object candidate, it examines if the "supporting relation" holds between already generated candidate and the new one. By checking supporting relation between objects, the system eliminates object candidates that are impossible to exist and estimates actual objects from parts seen in an image.

The "supporting relation" holds when an object can be considered to locate on another object and to be supported by it. Checking the "supporting relation" is carried out by examining whether to-be-supported regions of an object is almost included in supportable regions of another object. If so, the former object is regarded to be supported by the latter one. We present an example including a relation that "`desk is supported by floor`" in Fig. 4.

In case that two or more candidates can supports the new candidates, for example, a book is on the floor and at the same time the book is on the desk, we regards that the new candidate is supported by the upper

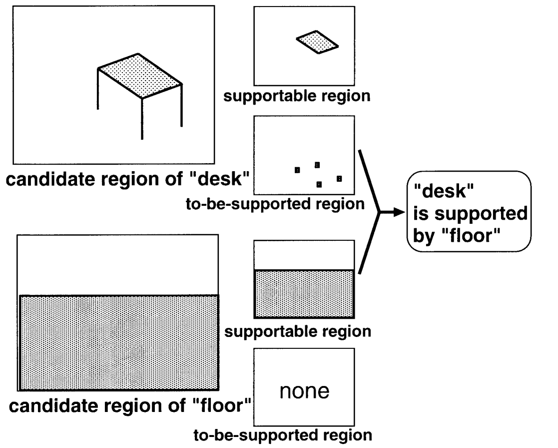


Fig. 4 Checking "supporting relation."

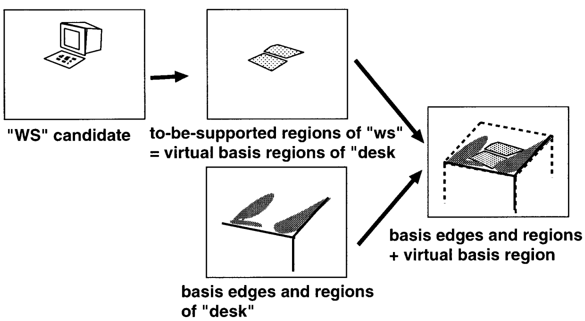


Fig. 5 Estimating a "desk" candidate that supports the "workstation" candidate.

candidate, that is, a book is on the desk.

If a candidate has no supporting relation, to-be-supported regions of the candidate are regarded as **"virtual basis regions"** and the system searches a new candidate with supportable regions including the virtual basis regions (Fig. 3). In short, "virtual basis regions" are regions that can be regarded as regions of supportable elements of a new candidate. Then, the system can detect a new candidate that couldn't be detected before. For example, when the system generates a "workstation (WS)" candidate with no supporting relation, it regards the to-be-supported region of "WS" as the virtual basis regions of a desk face element (Fig. 5). By this mechanism, the system recognizes an object occluded by another object.

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

3.3 Relational Knowledge

The system has "relational knowledge." They are descriptions about relative relation generally expected between two objects. It is used for computing confidence value of relation and expecting the region where own target object exists with a high possibility. It is represented by combination of "relation name," "source object's name" and "destination object's name." For example, "`on(book,desk)`" means "a book is usually on a desk." At present, we have three types of relations, "`on`," "`in_front_of`" and "`on_same_plane`." The system judges whether each relation is holding using information on supporting relations and relative location of objects in an image.

Confidence value of relation V_{re} is determined with a weighted sum of holding relations:

$$V_{re} = 1 - \exp\left(-h \sum_{i=1}^{r} C_i n_i\right) \tag{4}$$

where r is the total number of relations, n_i represents if relation i is holding, that is a value of 0 or 1, and h is a constant. We set 0.4 to h for the experiments. C_i is a weighting factor that represents the degree of significance of each relation. At present, it is set 1.0 for the relation "`on`" and 0.5 for two other relations.

Equation (4) includes exponential term so that the increase of V_{re} is much larger in case where the number of holding relations increases from one to two, compared to the case where it increases from five to six. Confidence value of relation V_{re} takes the value from 0 to 1. It is estimated by checking relational knowledge one by one. It can be regarded as the degree of naturalness of existence of a candidate in the scene.

3.4 Conflict Resolution

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

$$V = (V_{im} \times S' + V_{re} \times w)/(S' + w) \tag{5}$$

$$S' = \min(S, 2w) \tag{6}$$

where S is the number of pixels of candidate elements and w is a constant that controls balance between V_{im} and V_{re}. The weights are decided with taking the size of a candidate into account. If the number of pixel 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} for the candidates having the small number of pixels in the evaluation of the V. We set 2500 to w for the experiments.

As a result of comparison of V, the candidate with the highest value remains, and all other candidates are canceled. If the ratio of conflicting regions to basis regions and edges of the canceled candidates is less than 0.5, "candidate modification re-recognition" is carried out. In this re-recognition, the conflicting candidate is modified into a new candidate including no conflicting regions.

In the calculation of confidence value, some parameters appear. Even if these parameters are changed slightly, the result of conflict resolution does not change in case where one is correct and another is incorrect apparently in the image. In such case, the difference of their confidence values is large, so that slight change of parameters does not affect results. In case where the difference of confidence values is small, the result of conflict resolution may change. In that case, however, it is hard to distinguish which one is correct in the image.

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 [11] and applied it for our system.

4. Overview of the System

4.1 System Architecture

We designed the system based on "MORE" architecture we proposed in [3]. It is multi-agent-based architecture and constructed as an assembly of agents that recognize objects from an image in parallel and independently. It enables to recognize various different kinds of objects by adding agents. In our system, one agent consists of **a recognition module** (RM), **a communication module** (CM) and **candidate objects** (CO) (Fig. 6). In addition, the system has **a feature extraction module** (FE) that extracts image features from an input image. The processing flow among all the modules is message-driven.

RM recognizes only one kind of target objects by sending FE requests to extract image features. CM carries out cooperation among agents. It checks supporting relations to candidates generated by other agents and resolves conflict among the agents. Using supporting relation and relational knowledge every CM has, it estimates the region where own target object exists with a high possibility in the re-recognition stage and issues re-recognition requests to RM. RM and CM exist from the beginning, but CO doesn't exist before starting of the recognition. It is generated by CM, every time a new object candidate is found.

Besides agents, the system has FE, which makes an image pyramid and extracts straight edges and regions by requests from RM of each agent. Since FE plays an role as a subcontractor of image processing in RMs, after here we regard FE as being included in RMs and don't mention it in this paper.

4.2 Recognition Requests

CM sends recognition requests to RM, and RM starts recognition processing. There are six kinds of recognition requests (Table 2). One is an initial recognition request, and others belong to re-recognition requests. When each request is issued, in according to the size of a region to recognize, CM selects proper resolution level of the image.

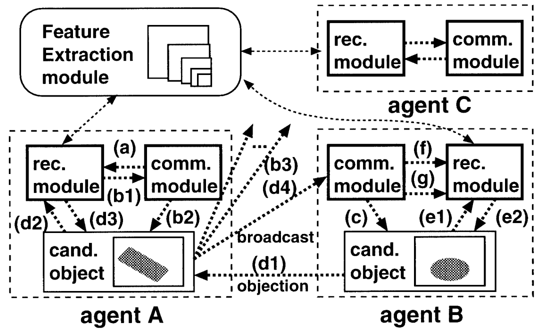


Fig. 6 System structure and flow of messages. (a) **initial recognition request** (b1) information of an new object candidate (b2) generation of a candidate object (b3) broadcasting information of a new candidate (c) notifying a new candidate of another agent (d1) objection message (d2) **modification request** (d3) information of a modified candidate (d4) cancellation message (e1) **renewal request** (e2) information of a renewed candidate (f) **supporting request** or **to-be-supported request** (g) **recognition request for vacant regions**.

- **Initial Recognition Request**
This request is issued at the beginning of processing. Recognition is carried out for a whole input image of the highest level.

- **Modification Recognition Request**
After conflict resolution, this request is issued in order to modify the region of a canceled candidate into a new region with no conflict. For the region of the canceled candidate in the proper level image, the RM of the same agent as CO of the canceled candidate tries to re-recognize a new candidate whose basis regions and edges don't overlap with basis regions and edges of the opponent candidate in the conflict resolution.

- **Renewal Recognition Request**
In case that a newly generated candidate makes no conflict with other candidates, the candidate is re-recognized in its proper level by this request.

- **Supporting Recognition Request**
If a new candidate except background objects is not supported by any other candidates, this request is issued in order to search a supportable candidate in the region under the new candidate in the proper level image. This request is issued in agents whose objects can support the new candidate according to relational knowledge.

- **To-be-supported Recognition Request**
When a new candidate with supportable regions is recognized, this request is issued. The supportable regions of the candidate are examined by agents whose object is supported by the candidate according to relational knowledge.

- **Recognition Request for Vacant Regions**

Table 2 Six kinds of recognition requests.

request type	region to recognize	resolution level	condition	kinds of target objects
initial	whole image	max (the lowest resolution)	none	all kinds
modification	original candidate	proper	not including basis regions and edges	same to original candidate
renewal	original candidate	proper	none	same to original candidate
supporting	the virtual basis region and its peripheral region	proper	a supporting region including the virtual basis region	objects described in relational knowledge
to-be-supported	supporting regions and its peripheral region	proper	a to-be-supported region being included in the supporting region of the source candidate of requests	objects described in relational knowledge
vacant regions	unrecognized regions	proper	none	all

In the end of processing, in case that there are regions where no candidates detected in an image, this request is issued as the final re-recognition request.

4.3 Flow of Recognition

As was noted before, the processing flow among all modules is message-driven. We describe the detail flow of messages and recognition requests in the case shown in Fig. 6.

(a) Initial Recognition
At first, CM sends RM an **initial recognition request**, and RM initiates *initial recognition* for the whole region of the highest level image with minimum resolution (Fig. 6 (a)). If no candidate is detected, RM executes *initial recognition* for next lower level image again.

(b) Generating Object Candidates
When a new candidate is detected at RM, its information is sent to CM (b1). After checking supporting relation, CM generates CO (b2), and CO broadcasts information about it for CMs of other agents (b3).

(c) Receiving Information of Candidates
If another CM receiving information of a new candidate founds conflict, the CM informs occurrence of the conflict to CO concerned with conflict with it (c).

(d) Conflict Resolution
CO send an objection message to CO of the other agent (d1), and COs concerned with the conflict carry out conflict resolution by comparing each confidence value of a candidate and relation. The winning CO remains, and the losing CO sends CM a **modification request** (d2), (d3). By the *modification request* RM re-recognizes the region of the losing candidate in the proper level image to modify its own region lest conflict occurs. If modification fails, the candidate is canceled, the CO broadcasts a cancellation message (d4) and it is terminated finally.

(e) Renewal of Object Candidates
If a new generated candidate doesn't make conflict, a **renewal request** is sent from CO to CM, and RM re-recognizes the region of the candidate in the proper level image (e1), (e2).

(f) Estimation of Candidate Region based on Supporting Relation
If CM receives information of a new candidate without supporting relation with any other candidates, CM sends RM a **supporting request** and RM searches a supportable candidate object for the unsupported candidate in the region under the candidate in the proper level image (f).

If CM has relational knowledge related to a new candidate sent from other CM, CM sends RM a **to-be-supported request** and RM examine supportable regions of the new candidate object in order to detect its own new candidate.

(g) Re-recognition for Vacant Region
If all modules of all the agents are in the state of waiting for a message and there is no message on communication lines, the system enters the final recognition stage. In the final recognition, if there are regions where no candidates detected in an image, a **recognition request for vacant regions** is sent (g). Re-recognition for these regions is carried out.

After the final recognition, the whole recognition of the system completes.

5. Recognition Results

We have implemented an experimental system for indoor images with six agents ("desk," "chair," "floor," "book," "pen," and "work station (WS)") on PC cluster system that consists of six PCs (Intel Celeron 450 MHz, memory 128 MB) using the PVM library [12]. In this system, each agent is implemented on each one PC.

5.1 Examples of Recognition Results

A sample indoor image (1280×960) in Fig. 7 includes a

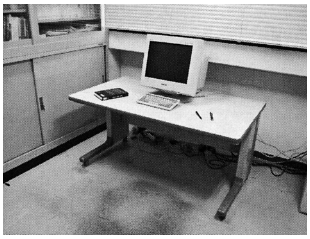


Fig. 7 An indoor sample image.

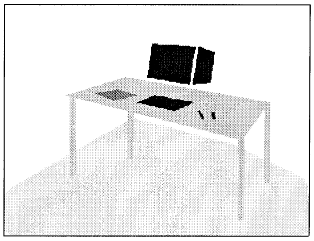


Fig. 8 Recognition result using multi-resolution images.

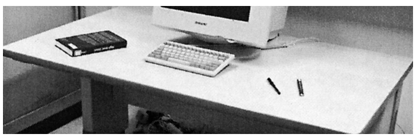


Fig. 9 The region of the desktop.

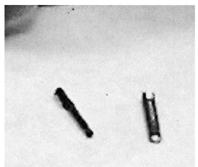


Fig. 10 The region of the pen candidate.

"book," a "pen" and a "WS" on a "desk." In the experiment by single-resolution recognition for the 320×240 image, only a "desk," "floor" and a "WS" were recognized correctly, but a "pen" and a "book" were not recognized.

In multi-resolution analysis, we used an image pyramid consisting five level images from level 0 to 4. Reduction ratio was 0.7, and size of the highest-level image (level 4) was 308×231. In the experiment, at first, in initial recognition for level-4 image a "desk," "floor" and a "WS" candidates were generated. Next, by relational knowledge of `on(book,desk)` and `on(pen,desk)`, "book" agent and "pen" agent initiated re-recognition for the region of "desk" and its peripheral region in level-3 image by a *to-be-supported request*, and generated a "book" and "pen" candidates, respectively (Fig. 9). After that, both candidates were re-recognized in level-0 image by a *renewal request* (Fig. 10). Finally, we obtained a result shown in Fig. 8. While this process, conflict between a keyboard part of a WS and a book occurred, and some parts of a bookshelf of left hand were recognized as a book and a pen candidate. However, they were canceled finally by conflict resolution and checking "supporting relation."

We show execution times in case of single-resolution for three sizes of the image and multi-resolution for the sample image (Fig. 7) in Table 3. With 640×480 image all objects were detected, but the execution time was about five times as long as one with 320×240 image. The system employing multi-resolution analysis could recognize all objects, and its execution time was only about twice as long as one with 320×240 image.

We show more complex indoor scene image in

Table 3 Execution time.

resolution	time[sec.]
320 × 240	8.3
640 × 480	38.3
1280 × 960	× (out of memory error)
multi-resolution	19.6

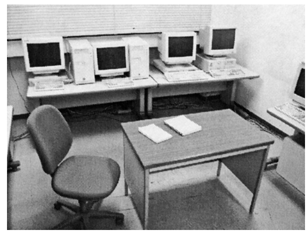


Fig. 11 A complex indoor sample image.

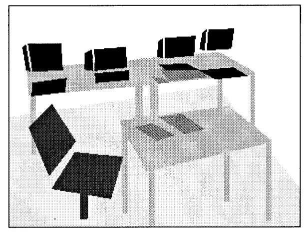


Fig. 12 Recognition result for Fig. 11.

Fig. 13 Another complex indoor sample image.

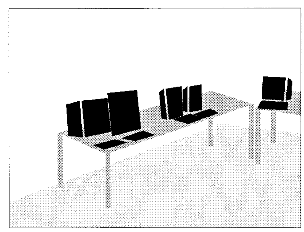


Fig. 14 Recognition result for Fig. 13.

Fig. 11 and its recognition result in Fig. 12. In the experiment, only with 320×240 image the system couldn't recognize four WSs on the back desks. However, it recognized such a complex image by forming five level resolution images.

Another sample image in Fig. 13 is also relatively complex. Its recognition result is shown in Fig. 14. In this experiment, one back WS was not recognized, but other five WSs could be recognized correctly. Especially, although three recognized WSs on the back were appeared as only small regions in the image, they could be recognized by multi-resolution analysis.

5.2 Experimental Results for 20 Images

We made experiments for 20 images including various indoor images from a simple image like Fig. 7 to a relatively complex image like Fig. 11 and Fig. 13. Figure 15 shows 12 images out of 20 samples images used in this experiments, and Fig. 16 shows recognition results for these 12 images in the multi-resolution recognition.

We divided the results into "almost correct," "half correct" and "almost incorrect" in terms of the recognition rates. The recognition rates for "almost correct," "half correct" and "almost incorrect" are 80%–100%, 30%–80% and 0%–30%, respectively. Table 4 shows the experimental results for 20 images in case of both the

Fig. 15 12 out of 20 images for experiments. Images in the upper row are quite simple, ones in the lower row are complex, and ones in the middle row have middle complexity.

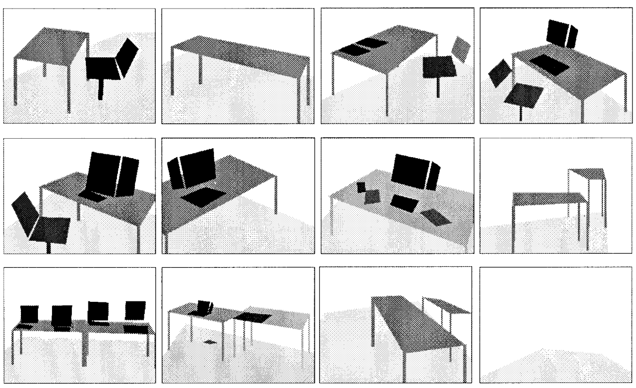


Fig. 16 Recognition results for above 12 images.

Table 4 Results for 20 images.

	almost correct	half correct	almost incorrect
single-resolution (320 × 240)	5	8	7
multi-resolution (5 level)	12	3	5

single-resolution recognition and the multi-resolution recognition.

In the single-resolution recognition, only five images were interpreted correctly, but in the multi-resolution recognition, 12 images were interpreted correctly. Both numbers of almost incorrect results were comparable. This is because these images were too complex to extract significant image features in the initial recognition stage, so that effective re-recognition couldn't be initiated.

6. Conclusion

In this paper, we applied multi-resolution analysis to a multi-agent-based image recognition system. We proposed and realized the system that can use a high-resolution image effectively without much extra processing time.

For future work, we plan to construct recognition modules by machine learning methods and introduce more effective cooperation mechanism to improve ability of individual recognition modules.

References

[1] Z. Li and L. Uhr, "Pyramid vision using key features to integrate image-driven bottom-up and model-driven top-down process," IEEE Trans. Syst., Man & Cybern., vol.17, no.2, pp.250–263, 1987.

[2] C.L. Tan and W.N. Martin, "An analysis of a distributed multiresolution vision system," Pattern Recognition, vol.22, no.3, pp.257–265, 1989.

[3] K. Yanai and K. Deguchi, "An architecture of object recognition system for various images based on multi-agent," Proc. 14th Int. Conf. of Pattern Recognition, vol.1, pp.278–281, 1998.

[4] K. Yanai and K. Deguchi, "Recognition of indoor images employing qualitative model fitting and supporting relation between objects," Proc. 15th Int. Conf. of Pattern Recognition, vol.1, pp.964–967, 2000.

[5] B. Draper, R. Collins, J. Brolio, A. Hanson, and E. Riseman, "The schema system," Int. J. Computer Vision, vol.3, no.2, pp.209–250, 1989.

[6] T. Matsuyama and V.S. Hwang, SIGMA: A knowledge-based aerial image understanding system, Plenum Press, New York, 1990.

[7] J.Y. Aloimonos, "Preface of special issue on purposive, qualitative, active vision," Computer Vision and Image Understanding, vol.56, no.1, pp.1–3, 1992.

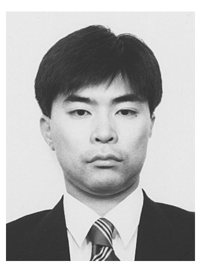
[8] L. Stark, "Functionality in object recognition," Computer Vision and Image Understanding, vol.62, no.2, pp.145–146, 1995.

[9] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Analy. & Mach. Intell., vol.5, no.2, pp.140–150, 1986.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," Int. J. Computer Vision, vol.1, no.4, pp.321–331, 1988.

[11] D. Kim and R. Nevatia, "Recognition and localization of generic objects for indoor navigation using functionality," Image and Vision Computing, vol.16, pp.729–743, 1998.

[12] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, PVM: Parallel Virtual Machine, The MIT Press, 1994.



Keiji Yanai received B.Eng. and M.Eng. degrees from University of Tokyo in 1995 and 1997, respectively. From 1997, he is a research associate of Department of Computer Science, the University of Electro-Communications. His recent research interests include an image understanding system and a multimedia database.

Koichiro Deguchi received B.Eng., M.Eng. and Dr.Eng. degrees from University of Tokyo in 1974, 1976 and 1980, respectively. From 1984 to 1987, he was an associate professor of Department of Information Engineering, Yamagata University, and from 1988 to 1998, he was an associate professor of Department of Mathematical Engineering and Information Physics, University of Tokyo. From 1998, he is a professor of Graduate School of Information Sciences, Tohoku University. His recent research interests include computer vision, signal processing and multi processor architecture for them.