An Image Understanding System for Various Images Based on Multi-Agent Architecture

Keiji Yanai

Department of Computer Science, 5 The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, JAPAN
yanai@cs.uec.ac.jp

Abstract

An image understanding system for real world images which has an ability to recognize various kinds of images is proposed. We propose a multi-agent architecture to integrate and cooperate object recognition modules for individual target objects. In our system, object candidates generated by different agents are integrated not only on the evaluations by each modules themselves but also on spatial relations among objects. By checking spatial relation, the agents also estimates actual objects from parts seen in the image. Such mechanisms are realized by autonomous cooperation among the agents, and the most reliable result is selected after the arbitration between them. We implemented an experimental system on PC cluster system, and achieved recognition for both indoor and outdoor images.

1. Introduction

An image understanding system for the real world must be able to recognize various kinds of images. We have been developing the system that recognizes artificial and natural objects in a single image[5]. Here, "recognition" means to obtain a category name of an object, for example "desk" or "chair". Therefore, in the recognition we consider here we can't assume that exact shapes of target objects are known in advance unlike conventional researches on object recognition.

To achieve such image understanding, we have to utilize not only shapes of objects but also relations between them as a recognition clue. We expect the knowledge about relations between objects which hold in normal situation to compensate for incomplete knowledge about shapes of objects. All objects except background object must be supported by other objects in the real world due to gravity of the earth, so by checking supporting relation we can estimate existence of objects and eliminate object candidates that are impossible to exist.

To realize such recognition we propose an architecture of an assembly of agents. In our architecture each agent consists of a recognition module to recognize a single kind of object and a communication module to communicate with other agents. In this paper, we describe the system architecture and experimental results on the system implemented on PC cluster system.

The conventional image understanding systems based on multi-agent architecture aimed at integration of multiple algorithms[3] or flexible use of relations among objects[2]. They needed consideration of interaction among agents one by one and complicated description of relations among objects. Therefore they usually restricted the target image. To construct such general recognition system, their architecture is not effective. In our architecture, we only prepare recognition modules for each single kind of object independently. Adding co-operation mechanism, that is communication module, to integrate them, we can implement the system easily.

2. Basic concept of our system

In our architecture the system is constructed as an assembly of agents that recognized objects from an image independently. By adding agents it enables to recognize different kinds of images. One agent consists of a recognition module and a communication module (Fig.1). Each agent runs in order to recognize as many objects as possible in the image.

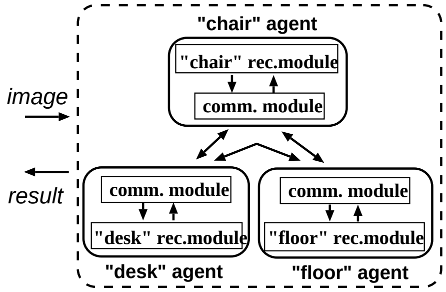

Figure 1. Basic structure.

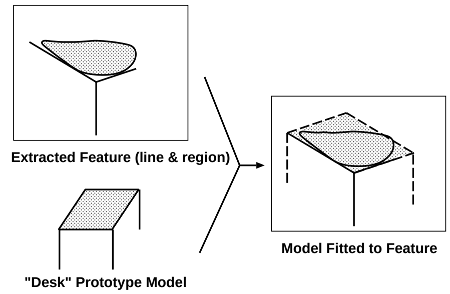

Figure 2. Estimation of "desk" region.

Then sometimes conflicts occur among agents, that is, two or more agents find different objects in the same region on the image. Then they negotiate with each other to resolve the conflict. Each agent also makes use of information of object candidates generated by other agent. For example, when "a car" is found, there possibly exists "road" under it. Such cooperative process is managed by communication modules, while recognition modules deal with input images directly.

3. Recognition module and Communication module

A **recognition module** recognizes only one kind of target object as regions in an input image, and generates an object candidate. In principle, a recognition module recognize objects in the simple way. The recognition strategy is quite different according to kinds of category of target objects. We distinguish object category by three kinds, artificial objects, natural objects and background objects.

For artificial bjects, we recognize by paying attention to the part that represents its function[4]. For example, region of sitting surface is paid attention for "chair" and desk surface is for "desk". For searching region of sitting surface, we extract approximately elliptic or four-sided regions by region segmentation algorithm, and for searching region of desk surface, we extract a parallelogram under which there are four vertical lines corresponding to desk legs by hough transformation. As a matter of course, in most case the perfect regions or lines cannot be extracted, so we need estimate region of sitting or desk surface from partial region or lines that can be extracted. We call the partial regions or lines "object clues". Then, we fit prototype model of chair or desk to partial region or lines (Fig.2). The prototype model represents essential structure of the objects. It is represented by polygons and straight line segments, and relative ratio of their size is variable within specified extent. Some prototype models are prepared for one category of object, and best fitting model is selected.

For natural objects, like "tree" or "mountain", using region segmentation based on texture and "snake" algorithm recognition modules extract regions.

For background objects, it has no specific visual shape like "road" or "floor", so we make use of region segmentation algorithm, and a recognition module fits polygonal region to extracted "object clue". This enable to estimate whole region of background object behind the artificial or natural objects.

As described above, a recognition module extract "object clues" basically by combination of conventional methods, for example, sobel filter, hough transformation, region growing, snake and so on. It estimates the whole region of the object candidate by fitting a prototype model to "object clues", and besides it estimates regions that can support other objects and regions that must be supported by an other object using prototype model for "checking supporting relation" described later.

Every time a recognition module finds an object candidate, it estimates evaluation score of it, then, sends the information about the regions and the evaluation score to the communication module.

A **communication module** carries out cooperations among agents. It keeps consistency to other agents to check other candidates each other. If a conflict occurs, it negotiates with the communication modules of agents concerning the conflict.

Figure 3. Examples of relational knowledge.

```
desk on floor
ws on desk
book on desk
desk taller_than chair
book smaller_than desk
```

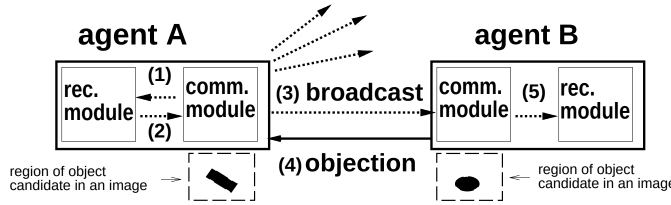

Figure 4. Flow of messages.

Every communication module has "relational knowledge". It is a description about relation generally expected between two objects, and it is represented by triplets of "source object's name", "relation name" and "destination object's name". We show some examples in Table 3. A communication module has only relational knowledge related to own target object. Using relational knowledge, a communication module estimates evaluation score about relation and regions where own target object exists with high possibility. It also checks supporting relation to candidates generated by other agents.

4. Recognition flow

The processing flow of all the modules is message-driven. We describe detail flow of messages in case of the example of Fig.4.

(1) An input image is sent to recognition modules of all agents. Then each recognition module starts recognition by the request from the communication module.
(2) Every time a recognition module finds an object candidate, it sends the information of the regions and its evaluation score to the communication module.
(3) The communication module broadcasts it for all other agents.
(4) Other agents examine if the broadcast is consistent with own object candidates. If not, the agent sends back an objection message. Then, conflict resolution is carried out between the communication modules concerned.
(5) When some relations exist between the broadcasted object candidate and a target object of the receiving agent, the communication module of the receiving agent estimates region where a target object is expected to exist. The communication module sends "conditional recognition request" to the recognition module.

In addition to this, our architecture has "revival mechanism". It revives an object candidate that was canceled once, when evaluation about relation is changed and the result of comparison becomes invalid. By this mechanism the system always keep object candidates that are consistent with each other.

If the modules of all the agents are in the state of waiting for a message and there is no message on communication lines, the whole recognition of the system have completed.

5. Cooperation among agents

5.1. Conflict resolution

"Conflict" means that there exists the region where two or more object candidates are assigned. When a conflict occurs, the concerning agents compare evaluation about object candidates and cancel one candidate. The evaluation for candidate is done by combination of evaluation score about shape and relation. In case of conflict between three or more objects, the agents resolve it between two of them.

Evaluation about shape for a generated candidate is estimated by each recognition module. In principle it scores a candidate in proportion of detected object clues to regions and lines expected from a prototype model.

Evaluation about relations is estimated by each communication module examining the number of relations realized between its own and other objects. The score is the summation of weights of applicable relations.

Two conflicting candidates are compared by their evaluation scores about shape at first. If difference between two scores is more than certain value, smaller one is canceled. Otherwise, scores about relation are compared, then smaller one is canceled. If both differences are small, temporary decision is made by comparing sizes of their regions.

5.2. Checking supporting relation

All objects except background objects must be supported by other objects in the real world due to gravity of the earth. According to this fundamental rule every time an object candidate is generated by a recognition module, a communication module examine if "supporting relation" holds between already generated candidate and it.

"Supporting relation" holds when the object can be considered to be located on another objects and to be supported by it. Checking "supporting relation" is carried out by examining if the region of an object that must be supported is almost included in the region of another object that can support other objects for two objects. If so, the former object is regarded to be supported by the latter object. We present examples including "`book is supported by desk`" in Fig.5.

If an object candidate have no supporting relation, its agent sends "supporting request" to agents of objects expected to support it. Objects expected to support it are all background objects and objects with which its agent has relational knowledge "on". Then, the communication module of the agent receiving it sends "conditional recognition request" to its recognition module, and it carries out re-recognition processing.

5.3. Recognition using relational knowledge

Information on object candidates is broadcasted for reporting it to all the other agents. At this time by integrating received information on candidates and relational knowledge, the agents sometimes can expect the region where its own objects exist. For example, "desk" agent generates an object candidate, and "book" agent gets the information. Using relational knowledge "`book on desk`" it can estimate that a square region over "desk" region is "book".

The communication module of each agent examines if it has some relation to received object candidates as soon as it receives. If some relation is found, the communication module sends it "conditional recognition request" to the recognition module.

6. Implementation and experiment

We have implemented an experimental system with 12 agents ("desk", "chair", "workstation", "wall", "floor", "house plant", "book", "cup", "road", "car", "sky" and "tree") on PC cluster system which consists of 8 personal computers (Intel Celeron 300A) using PVM library[1].

In the experiment for an image of indoor scene shown in Fig.6 fourteen object candidates (two "books", three "desks", a "chair", a "workstation", two "floors", "wall", a "house plant", a "cup" , "tree" and "sky") were generated by the respective agents. During this recognition conflict resolution occurred thirty-five times, compatible occurred twenty-eight times, cancellation occurred seven times, and revival occurred one times. Finally eight objects remained as shown in Fig.7. A conflict occurred between desk candidate no.1 (Fig.8) and chair candidate (Fig.9), and desk no.1 was canceled by comparison of the evaluation scores about shape and relation.

There were region overlaps between floor (Fig.11) and chair and between floor and desk. But "supporting relation" holds between two, so they are regarded as compatible. A region overlaps also occurred between desk and workstation, between desk and book and so on. However, because between desk and workstation and between desk and book "supporting relation" holds, they are regarded as compatible, too.

Checking "supporting relation" is carried out between workstation and desk and between floor and desk. Since workstation candidate was found, desk no.1 (Fig.8) was extended to desk no.2 to support the workstation candidate (Fig.10). In the same way floor no.1 (Fig.11) was extended to floor no.2 to support desk candidate no.2.

In this experiment recognition using another object candidate and relational knowledge was carried out. "Book" agent recognized book candidates using the information of a desk candidate and relational knowledge "`book on desk`". Thus, though the recognition module simply regarded parallelograms as a book, a book could be recognized.

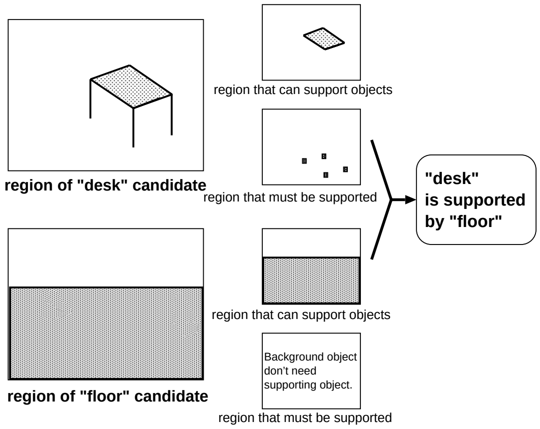

Figure 5. Checking supporting relation.

Figure 6. Indoor image.

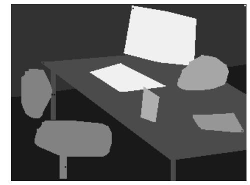

Figure 7. Recognition result.

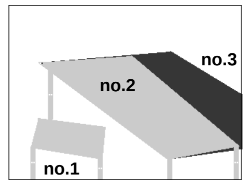

Figure 8. Candidates of "desk".

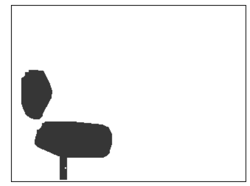

Figure 9. A candidate of "chair".

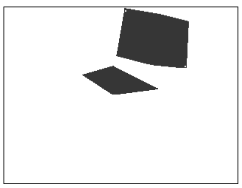

Figure 10. A candidate of "WS".

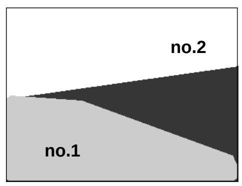

Figure 11. Candidates of "floor".

Figure 12. Outdoor image.

Figure 13. Recognition result.

In the experiment for the image of outdoor scene shown in Fig.12 five candidates ("sky", "a car", "road", "tree" and "wall") were generated. Conflict occurred once between "sky" and "wall", and "wall" was canceled by comparing evaluation score about relation. "Wall" has no relational knowledge to other object candidates, while "sky" has relation knowledge "`coexists_with`" to "road" and "tree". Finally we obtain the result shown in Fig.13.

7. Conclusion and future study

In this paper we described an architecture of image understanding system based on multi-agent architecture. It is constructed as an assembly of agents for individual objects. Each agent has recognition and communication modules. The system recognizes both outdoor and indoor images by cooperation among agents using relational knowledge.

For future works, we will study how to evaluate object candidates, more effective cooperation mechanism and implementation of each recognition module.

References

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

[2] T. Matsuyama and Y. Ohta eds. Special issue on "parallel, distributed and cooperative image understanding". *Journal of Japanese Society for Artificial Intelligence*, 9(5):629–643, 1994. (in Japanese).

[3] Y. Ohta, M. Watanabe, and Y. Sumi. Approaches to parallel computer vision. *IEICE Transactions on Information and Systems*, E74(2):417–426, 1991.

[4] L. Stark. Functionality in object recognition. *Computer Vision and Image Understanding*, 62(2):145–146, 1995.

[5] K. Yanai and K. Deguchi. An architecture of object recognition system for various images based on multi-agent. In *Proc. of 14th International Conference of Pattern Recognition*, volume 1, pages 278–281, 1998.