MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

A Modularized Vision System using a Distributed Cooperative Architecture

Tomoyuki Hamada
Mechanical Engineering Research Laboratory, Hitachi, Ltd.
502 Kandatsu, Tsuchiura, Ibaraki 300, Japan
E-mail: *hama@merl.hitachi.co.jp*

Abstract

In this paper, a new method for designing modularized vision systems using a distributed cooperative architectures was introduced. In this method, the target of vision is defined in advance using a set of parameters, the relations between which are determined by primitive function-based vision modules. The parameters themselves are determined simultaneously by running these vision modules under a cooperation mechanism. Using this architecture, three types of vision system were actually implemented and tested. The results experimentally verified that vision systems designed using the proposed method work successfully.

1. Introduction

There are high hopes for vision systems that generate a "workspace maps[1][2]" for use in rationalizing production, construction, and maintenance work. The workspace map is a dynamic data set describing three-dimensional arrangement of objects in the workspace, and is used for intelligent navigation of mobile robots, strategic planning of object handling, and dexterous manipulation of robot hands.

From a practical point of view, the vision system construction is determined by its application to the practical systems. That is, the number of imaging devices, the type of imaging devices, and their geometric configuration are designed considering the required accuracy, the degree-of-freedom of the object location, and the application task. Therefore, a number of specific vision systems have been developed for every practical applications. However, it is not efficient to develop each new vision system in a from-scratch manner. A constructive design method is needed that will allow designers to reuse resources developed in other application systems.

This paper proposes a new method for designing modularized vision systems using a distributed cooperative architecture. With this method, the target of vision is defined in advance using a set of parameters, the relations between which are determined by primitive function-based vision modules. The parameters themselves are determined simultaneously by running these vision modules under a cooperation mechanism. Using this method, any desired vision capability can be obtained, simply by combining these primitive vision modules.

In related work, Henderson has provided a programming paradigm for sensory information processing (the logical sensor concept[3]), and Brooks has proposed a subsumption architecture for evolutionary development of visuomotor systems[4]. However, in these works, the information fusion process is not made clearly in constructive style.

2. Distributed Cooperative Architecture

The architecture of vision systems can be categorized into hierarchical architectures[5] and parallel architectures[4]. Though the former are good for obtaining the high-level information of the environmental map, it is generally difficult to divide such systems into separate reusable vision modules. On the other hand, the latter are good for modularization, but it is generally difficult to build high-level information from their simple vision functions. Considering this problem, we introduced a distributed cooperative architecture which is capable of handling high-level information and has good modularity at the same time (Fig. 1).

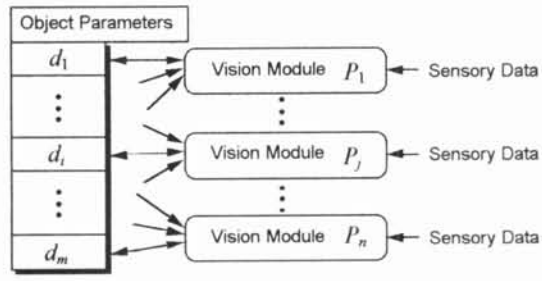

Fig.1 Distributed cooperative architecture.

The basic goal of the vision calculation can be considered as determining a set of parameter values about the object. In general, these parameters are not directly measurable, but intermediate data or feature values which are function of the object parameters are measurable. However, it is usually an ill-posed problem to determine the object parameters from these intermediate data. Then, the vision problem is formulated as non-linear optimization problem. That is, defining square error

$$L(\hat{\mathbf{d}}) = \sum_j ||G_j(\hat{\mathbf{d}}) - G_j(\mathbf{d})||^2 ,$$

estimate of object parameter $\hat{\mathbf{d}} = (\hat{d}_1,\ldots,\hat{d}_m)^t$ is obtained by minimizing L, where $\mathbf{d} = (d_1,\ldots,d_m)^t$ is the actual object parameter, and G_j (j=1...n) is the function which gives the j th intermediate data from the object parameters.

The proposed architecture is designed to solve this optimization problem by using a set of primitive vision modules P_j (j=1...n). That is, if a set of estimation function $E_j(g_j, \hat{g}_j)$ exists and they give $-\partial L / \partial \hat{g}_j$ or corresponding value, the optimal estimate of the object parameters are expected to be obtained by parallel iterative calculation:

$$\hat{\mathbf{d}}^{(t+1)} = \hat{\mathbf{d}}^{(t)} + E_j(g_j, \hat{g}_j^{(t)})$$
$$\hat{g}_j^{(t)} = G_j(\hat{\mathbf{d}}^{(t)})$$
$$(j=1...n).$$

Then, we design the primitive vision module P_j to execute the calculation for the j th parameter (Fig. 2). Since calculation of P_j is independent from other calculation except for shared referring of the object parameter $\hat{\mathbf{d}}$, we can design the internal procedure of P_j independently from other modules, and also the consistency between the vision modules which makes the system capable to produce a high-level information is maintained by sharing the same set of object parameters.

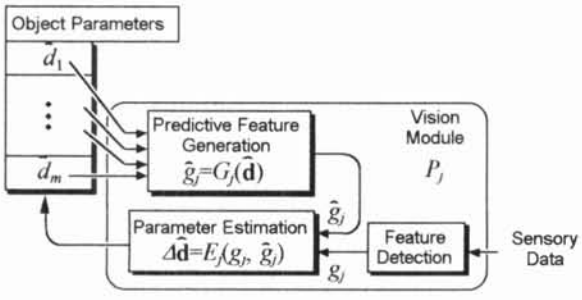

Fig.2 Recursive parameter estimation in a vision module.

3. Practical Implementations and Experiments

In order to verify the proposed method, we actually implemented three types of vision systems, and conducted experiments using them.

3.1 Binocular Object Tracking System

In order to track an object in three-dimensional space, we generally need an information fusion process between two images taken from different view points. However, by using the proposed method, a three-dimensional tracking system can be built simply by combining a pair of two-dimensional image-fitting modules (Fig. 3).

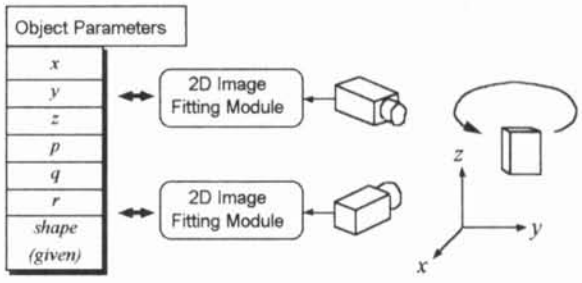

Fig. 3 Binocular object tracking system.

For this implementation, we define position (x, y, z) and orientation (p, q, r) as the object parameters, and prepare a pair of two-dimensional image-fitting modules. We use two edge-patterns of object images taken from different view points as the intermediate data g_1, g_2. In order to build the estimation function, we introduced the pattern kinetics matching method [6]. This method uses imaginary two-dimensional gravitation to fit the predictive edge pattern onto the detected edge pattern. Using the force vector $\mathbf{F}$ of this gravitation, the estimation function for the object position is given by

$$E_{Pos}(g,\hat{g}) = R_v sign(\mathbf{F})$$

where $sign(\mathbf{x})$ replaces each component of the vector $\mathbf{x}$ with an infinitesimal value with the same sign as the original value, R_v is coordinate transformation matrix from the view point coordinate to the world coordinate. $\mathbf{F}$ is given by

$$\mathbf{F} = \int_S \hat{g} \nabla u_g ds$$

where u_g is the two-dimensional potential field generated from the detected edge-pattern g.

In the same way, the estimation function for the object orientation is given by

$$E_{Ori}(g,\hat{g}) = R_v sign(\mathbf{N})$$

using moment vector $\mathbf{N}$ caused by the imaginary gravitation. $\mathbf{N}$ is given by

$$\mathbf{N} = \int_S \hat{g}(\mathbf{r} - \mathbf{r}_0) \times \nabla u_g ds .$$

Using these estimation functions, the calculation procedure of the j th vision module is defined as the following:

$$P_j : \quad \hat{\mathbf{d}}_{Pos}^{(t+1)} = \hat{\mathbf{d}}_{Pos}^{(t)} + E_{Pos,j}(g_j, \hat{g}_j^{(t)})$$
$$\hat{\mathbf{d}}_{Ori}^{(t+1)} = \hat{\mathbf{d}}_{Ori}^{(t)} * E_{Ori,j}(g_j, \hat{g}_j^{(t)})$$
$$\hat{g}_j^{(t)} = G_j(\hat{\mathbf{d}}_{Pos}^{(t)}, \hat{\mathbf{d}}_{Ori}^{(t)})$$

where operator $*$ produces a composed orientation vector of the two orientation vectors. Schematic illustration of this procedure is shown in Fig. 4.

Each fitting module is an independent vision system with a single-eyed camera, and estimates one two-dimensional projection of the full three-dimensional position and orientation. However, these components are fused to generate three-dimensional data in the simultaneous estimation process that takes place as the two single-eyed vision modules run in parallel.

Figure 5 shows the results of a three-dimensional object tracking test. In this experiment, the target object moved around continuously in three-dimensional space, and was tracked by two cameras.

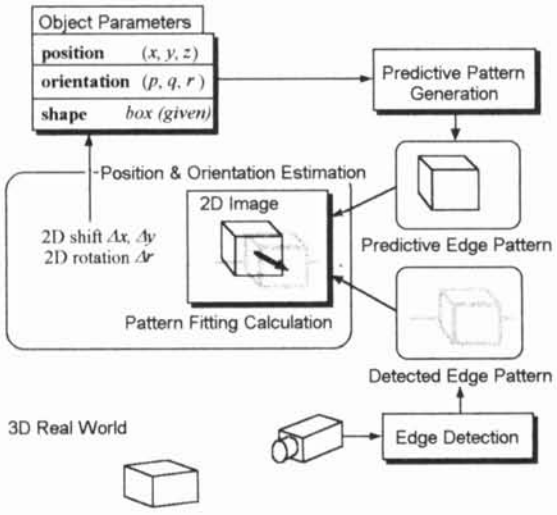

Fig. 4 Two-dimensional pattern fitting module.

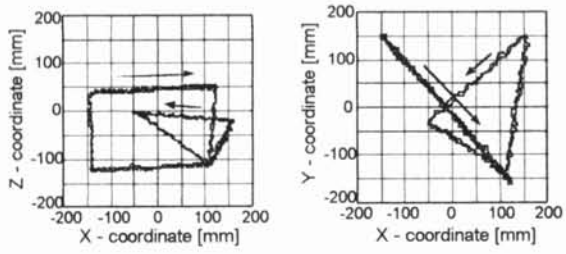

Fig. 5 Tracking result of three dimensional object motion.

3.2 Single-eyed Object Tracking System with Constraint

If the target object's motion is constrained to a plane, its three-dimensional position and orientation can be determined using single camera and the planar-motion constraint. In this implementation, we replaced one of the image fitting modules of the binocular object tracking system described above with a constraint module, as shown in Fig. 6. The calculation procedure of the constraint module is given by the following formula;

$$P_c: \quad \hat{\mathbf{d}}_{Pos}^{(t+1)} = \hat{\mathbf{d}}_{Pos}^{(t)} - \mathbf{n}\cdot(\hat{\mathbf{d}}_{Pos}^{(t)} - \mathbf{c}_0)\mathbf{n}$$

$$\hat{\mathbf{d}}_{Ori}^{(t+1)} = ||\hat{\mathbf{d}}_{Ori}^{(t)}||\,\mathbf{n}$$

where the plane is defined by norm vector $\mathbf{n}$ and a point $\mathbf{c}_0$ on that plane, and it is assumed that the object orientation is constrained so that the orientation vector $\mathbf{d}_{Ori}$ has the same direction as $\mathbf{n}$.

We applied this single-eyed tracking system to a mobile robot navigation system. In this case, given the location of a "landmark" object, its measured position and orientation relative to the camera are translated to produce the robot location, and are thereby used for navigation. Fig. 7 shows the results of a navigation experiment, in which a waving motion trajectory can be seen. (This trajectory is generated when the four legged robot moves its legs.)

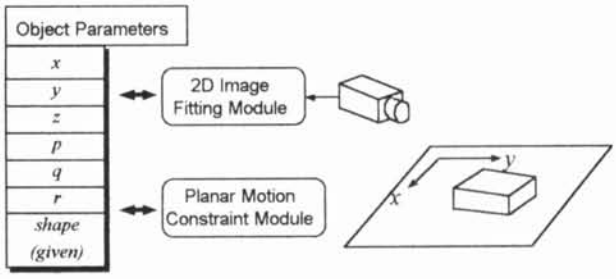

Fig. 6 Single-eyed object tracking system with planar motion constraint.

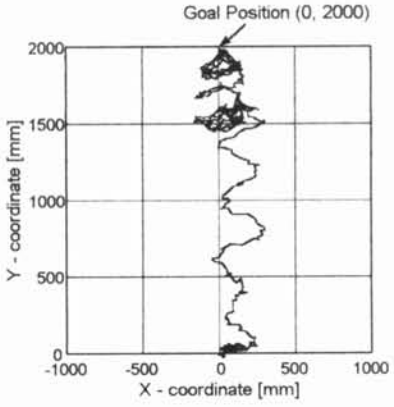

Fig. 7 Mobile robot navigation result.

3.3 Object Recognition System

In tracking systems, the object shape is assumed to be given. However, shapes can also be estimated by simultaneous calculation. In this implementation, we added a shape parameter d_{sh} and a shape recognition module to the single-eyed tracking system of section 3.2 (Fig. 8). The calculation procedure of this module is formulated as the following;

$$P_r: \hat{d}_{Sh}^{(t+1)} = E_{Sh}(g_j, \hat{\mathbf{d}}_{Pos}^{(t)}, \hat{\mathbf{d}}_{Ori}^{(t)})$$

where E_{Sh} is a search function which compares the object image g_j with a set of given shape models using the

temporal estimate of the position $\hat{\mathbf{d}}_{Pos}$ and orientation $\hat{\mathbf{d}}_{On}$, and determines the best matching shape.

The object recognition algorithm usually becomes a multi-searching program[7][8]. That is, the program has to be designed to find the best matching shape and the best fit position and orientation of the shape at the same time. However, the shape recognition module shown above is a simple pattern matching algorithm. The object shape, position, and orientation are determined simultaneously by running this module and the image fitting modules in parallel.

In the system implemented, an approximate object position is given in the initial values of the object parameters, and the system estimates the object shape, its exact position and orientation. Figure 9 shows the result of this object recognition experiment. In this figure, the recognized objects are indicated by the wire frames superimposed on them.

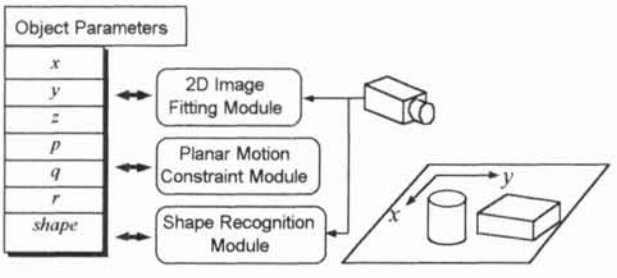

Fig. 8 Object recognition system.

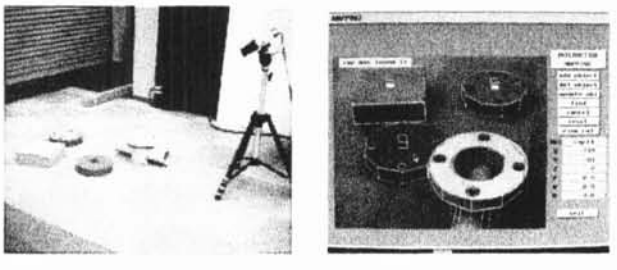

Fig. 9 Object recognition results.

4. Concluding Remarks

In this paper, a new method for designing modularized vision systems using a distributed cooperative architectures was introduced. Using this architecture, three types of vision system were actually implemented and tested. The results experimentally verified that vision systems designed using the proposed method work successfully.

Although many practical vision systems have been developed, there has been no way to reuse the resources developed in these systems, because each systems has been designed in a way specific to the individual application. By using our proposed method, one can design a vision system as a combination of reusable modules. As a result, we can extend system capability progressively by simply adding new modules.

However, this method requires that an appropriate estimation function should exist for every vision modules, and it is currently not clear that the estimation function can be built for any vision problem. In addition, we should consider the problem of the local minimum in the optimization process. These will be the future works.

Acknowledgments

This study was performed through the support provided by the Special Coordination Funds of the Science and Technology Agency of the Japanese Government.

References

[1] T. Hamada, K. Kamejima, and I. Takeuchi, "Dynamic Work Space Model Matching for Interactive Robot Operation," *Proc. IEEE Int. Workshop on Industrial Application of Machine Intelligence and Vision (MIV'89)*, Apr. 1989, pp. 82-87.

[2] T. Hamada, K. Kamejima, and I. Takeuchi, "Image Based Operation for Human-Robot Interaction," *IEEE Control Systems Magazine*, Vol. 10 No. 6, Oct. 1990, pp. 24-25.

[3] T. C. Henderson and E. Shilcrat, "Logical Sensor System," *Journal of Robotic Systems*, Vol.1 No.2, 1984, pp.169-193.

[4] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2 No. 1, Mar. 1986, pp. 14-23.

[5] D.Marr: *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H.Freeman & Co, San Francisco, 1982.

[6] K. Kamejima, Y. C. Ogawa, and Y. Nakano, "A Fast Algorithm for Approximating 2D Diffusion Equation with Application to Pattern Detection in Random Image Fields," *Proc. IMACS/IFAC Int. Symp. on Modeling and Simulation of Distributed Parameter Systems*, Oct. 1987, pp. 149-156.

[7] K. Ikeuchi, "Precompiling a geometrical model into an interpretation tree for object recognition in bin-picking tasks," *Proc. Image Understanding Workshop, DARPA Information Science and Technology Office*, 1987, pp.321-339.

[8] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, 3, 3, 1984, pp.3-35.