Coordination of an Uncalibrated 3-D Visuo-Motor System Based on Multiple Self-Organizing Maps

Min HAN**, Nobuhiro OKADA** and Eiji KONDO**

This paper presents a method to coordinate an uncalibrated visuo-motor system in a 3D space. In order to handle spaces occluded by obstacles, a 3-camera system and two related self-organizing maps (SOMs) are employed. The self-organizing maps are directly connected to the camera system and are trained to perform position control. Based on the visibility of targets given in the workspace, an appropriate map is adopted. The maps determine the joint angles of the manipulator which make the end effector reach the targets precisely, and make the manipulator take obstacle-free poses. The proposed learning method ensures that the manipulator moves smoothly and consistently in whole workspace even though we use two maps to control it. In our visuo-motor system, neither any priori knowledge about the manipulator nor the camera parameters is required. In addition, the system is robust to change in its geometry. Simulation results are presented to demonstrate the effectiveness of the proposed method and the robustness of the system.

Key Words: Neural Network, Manipulator, Learning, Positioning, Visuo-Motor System, Self-Organizing Map

1. Introduction

Vision guide for a robotic manipulator has been one of the major research issues for more than three decades. Coordination schemes of visuo-motor systems can be classified on the basis of the knowledge we have about the manipulator kinematics and about the camera parameters. A number of systems that deal with unknown robot kinematics or unknown camera parameters have been proposed in many literatures. Visuo-motor models are either estimated analytically during the execution of the task on-line or learned off-line prior to the execution of the task. Artificial neural techniques can be used to learn the non-linear relationships between features in the images and manipulator joint angles\(^{(1),(2)}\). In Ref. (1), Miller et al. proposed a neural network based on the learning control system, where a cerebeller model arithmetic computer memory is employed for the learning. In Ref. (2), an approach used to train an uncalibrated industrial robot is proposed. A neural network provides the estimate of the pose of the target in the manipulator coordinate frame. The pose is then used to guide the robot to grasp an object. However, a supervisor is needed in such a system.

Self-organizing feature maps based on the Kohonen\(^{(3)}\) algorithm are important unsupervised artificial neural network models that have shown great potential in application fields such as motor control, pattern recognition, and optimization, and have also provided insights into how mammalian brains are organized\(^{(4),(5)}\). During the past years it has been demonstrated that the self-organizing maps can solve the inverse kinematics problem for visuo-motor control\(^{(6)-(10)}\). In Refs. (6) and (7), arm movements were devised by tracking an image target. The correlation between an image-defined error and the joint movement servoing this error to zero was learned on-line using self-organizing algorithm. In their studies multiple neural maps were combined to simplify neural learning. In Refs. (8) and (9), they used a three-dimensional lattice to learn the nonlinear transformation which specifies the joint angles that takes the tip of a 3-DOF robot arm to the target point defined by retinal coordinates provided by two cameras. But in all of these studies, they solved the visuo-motor coordination problems with non-redundant manipulators in an environment without obstacles. However, obstacle avoidance problems are important for the robot...
manipulators while working in a real environment. In Ref. (10), using the TRN model, Zeller et al. showed a motion planning for a non-redundant manipulator to avoid collision obstacle in a cluttered environment. A locally optimized path can be determined by minimizing the Euclidean distance from the current position to a given target position. Collision-checking was completed not in self-organizing learning but in the path planning afterwards. In contrast to these existing methods, our self-organizing learning is not only to position the end effector precisely but also to ensure the manipulator acquiring obstacle-free poses. We intend to realize coordination for a visuo-motor system with a redundant manipulator in a cluttered environment. The redundancy is used to make the manipulator with obstacle-free poses and high manipulability.

In the previous researches, Zha et al. used a self-organizing map to realize coordination for a visuo-motor system in an environment that includes obstacles (11). However, collisions between the links of a manipulator and obstacles were not well considered. Okada et al. introduced a potential field to avoid collisions between a manipulator and obstacles (12). However, the visuo-motor system is only considered to move in a 2D space. Han et al. realized collision avoidance for the visuo-motor system in a 3D space (13). However, the occlusion problem was not solved effectively.

Vision systems are generally classified by the number of cameras, camera configurations, the level of calibration and some pre-knowledge about a scene. Binocular, stand-alone configuration is a commonly used configuration. Compared to the eye-in-hand stereo configuration, it allows for a wide field of view that makes it easy to observe both the manipulator and the targets simultaneously. Such a vision system was employed in Ref. (13). However, this vision system is vulnerable to handle parts occluded by obstacles in a cluttered environment. In Ref. (13), we treated the spaces occluded by obstacles in the image spaces as unreachable spaces for the manipulator, so the workspace was obviously restricted.

In order to handle the occlusion problem, we develop a visuo-motor system with two related self-organizing maps and a redundant camera system in this paper. The self-organizing maps are directly connected to the cameras and learned to perform motion control. Based on the visibility of targets given in the workspace, a more appropriate map is adopted. The map outputs a set of joint angle commands which make the manipulator reach the targets with obstacle-free poses. The proposed learning algorithm ensures that the manipulator moves smoothly and consistently in whole workspace no matter which map is used to control the manipulator. The advantages of the proposed method in this paper over other existing methods are: (1) By employing two maps alternately, the system overcomes the occlusion problems in cluttered enviroments. The cooperative and complementarity of two maps make the manipulator move in whole workspace consistently. (2) In our self-organizing learning procedure, the visuo-motor system is learned not only to position the end effector precisely but also to ensure the manipulator acquiring obstacle-free poses.

The paper is organized as follows. First, outline of our visuo-motor system is introduced in section 2. The learning procedure is discussed in section 3. Moreover, simulation results and an experimental system are shown in section 4. Finally, conclusions are given in section 5.

2. Our Visuo-Motor System

Our visuo-motor system is illustrated in Fig. 1. The system contains:
- A 4-DOF redundant manipulator moving in a 3D space
- Three CCD cameras
- Two related self-organizing maps

The CCD cameras are used to know the positions of the targets, the locations of the end effector and the poses of the manipulator. They also acquire information about obstacles. Based on visual information provided by the cameras, the self-organizing maps learn projections that convert the image vectors of targets in the image spaces into joint angles vectors of the manipulator. The manipulator is commanded by a set of joint angle commands which are the output of the self-organizing maps.

Although a stereo camera system can provide complete 3D information, spaces occluded by obstacles can-
not be dealt with well in our previous works. In this paper, a redundant camera system is presented to deal with the occlusion problem. Two of the cameras are arranged to see the workspace from the side, and the third camera observes from the top. The valid workspace is increased obviously by adding the third camera. Consequently, two related self-organizing maps are employed in our system. As shown in Fig. 1, the projections of a target point \( u_t \) in camera1, camera2 and camera3 are \((u_{R}, v_{R})\), \((u_{L}, v_{L})\) and \((u_{T}, v_{T})\) respectively. A pair of image coordinates of the camera1 \((u_{R}, v_{R})\) and camera3 \((u_{T}, v_{T})\) is combined into a 4 dimensional vector \((u_{R}, v_{R}, u_{T}, v_{T})\) which is used as the input of the map1. A pair of the image coordinates of camera2 \((u_{L}, v_{L})\) and camera3 \((u_{T}, v_{T})\) is combined into the input of the map2. Since the valid workspaces of two maps are different from the visible space of camera1 or camera2, two maps are used alternately.

Besides the number of movable joints, no further information about the manipulator and cameras will be used in our visuo-motor system.

2. The self-organizing maps

As shown in Fig. 2, each self-organizing map consists of neurons, which are distributed in the image spaces of the cameras. Each neuron \( N_i \) has 4 parameters.

- \( W_i \): Position of the neurons in two image spaces.
- \( J_i \): Jacobi matrix from the joint space to the image spaces.
- \( \theta_i \): Joint angle of the manipulator at \( W_i \).
- \( \xi_i \): Gradient vector of the evaluation function \( H \).

When a target \( u_t \) is given in the workspace, map1 or map2 is chosen based on which camera can see the target. In the chosen map, the neuron which \( W_i \) is the nearest to the projection of the target is chosen. The joint angles \( \theta^{\text{out}} \), which conduct the manipulator end effector to the target, are calculated obeying following linear equation (1).

\[
\theta^{\text{out}} = \theta_i + J_i^\dagger (u_t - W_i)
\]  

Where, \( J_i^\dagger \) is a pseudoinverse matrix of \( J \).

In the actual system, weighted sum of outputs from plural neurons around the target is used instead of Eq. (1).

\[
\theta^{\text{out}} = \frac{\sum g_n (\theta_i + J_i^\dagger (u_t - W_i))}{\sum g_n}
\]

Where, \( g_n \) is the weight defined by the following equation.

\[
g_n = \begin{cases} \exp(-n/\lambda), & \text{for } \exp(-n/\lambda) > \varepsilon \\ 0, & \text{for } \exp(-n/\lambda) \leq \varepsilon \end{cases}
\]

Where, \( n \) is the order of the neuron determined according to the distance between the neuron and the target. It has a large value for the neuron that is near to the target, and has a small value for the neuron that is far from the target. The symbols \( \lambda \) and \( \varepsilon \) are values to define neuron numbers that can affect \( \theta^{\text{out}} \).

3. Procedure of the Self-Organizing Learning

Two maps are employed in our system. If they are learned separately, the outputs of them are different even for the same target in the workspace. This will result in that the manipulator moves inconsistently when it is driven from one map’s valid workspace to another one’s valid workspace. In the learning algorithm, this problem has to be solved effectively. The problem also can be described as: the algorithm should guarantee that the manipulator moves smoothly and consistently in whole workspace no matter which self-organizing map outputs joint angles for it.
The learning procedure illustrated in Fig. 3. While a target position $u_t$ is presented randomly in the workspace, the cameras see the target. Depending upon which camera (camera1 or camera2) can see the targets (camera3 can always see all the workspace), map1 or map2 will be learned. While both maps learn for a common visible target, the maps do not learn separately. In this case, only one self-organizing map is chosen to determine the joint angles $\theta^\text{out}$ of the manipulator for the target. The manipulator is driven by $\theta^\text{out}$, and the cameras obtain different end effector positions $v$ in map1 and map2 respectively. Then the map1 and map2 correct the parameters by using $u_t$, $\theta^\text{out}$ and $v$ as the learning algorithm described in the following subsection. This learning procedure results in that in the end of learning the neurons of two maps possess the similar value of $\theta^\text{out}$, $\xi$ and different $W$, $J$ for the targets given in the common visible spaces. Thereby, the outputs from either map1 or map2 will ensure the manipulator has the same pose. This means: while a target is given in common visible parts, the output from either map1 or map2 does not result in an obvious change of the manipulator pose. In addition, the assignment of similar joint angles to adjacent target point is, in fact, one of the main features of the learning algorithm of the SOM. By the construction of a map between inputs in the image space and the neural net, learning algorithm makes sure that adjacent target points always activate adjacent neurons in the network. The learning forces adjacent neurons to adapt their output towards similar values. Therefore, at the end of the learning phase the output values will vary smoothly from a neuron to another neuron. Both features bring about a continuous and smooth transformation from the input image spaces of target points to the output space of joint angles. According to such a learning algorithm, the maps guarantee smooth and consistent movements of the manipulator in whole workspace.

3.1 Evaluation function

For redundant manipulators, finding the inverse kinematics mapping for a given end effector position is hard because this is an ill-posed problem in the sense that many solutions are possible. For two adjacent target points an algorithm may select two completely different joint angle configurations when there are redundant degrees of freedom. In our system, we introduce two evaluation functions to eliminate the under-determination of the manipulator control. The joint angle configuration, which optimizes the evaluation functions in addition to reaching the target point, will be selected. One evaluation function is $H_M$ to obtain high manipulability Eq. (3), the other one is $H_0$ to get obstacle-free poses Eq. (4). A general evaluation function $H$ is defined by the weighted sum of $H_M$ and $H_0$ as Eq. (5).

$$H_0 = \sum \left(1.0 - \frac{D_0}{d + D_0}\right)$$

$$H = \alpha_1 H_M + \alpha_2 H_0$$

Where, $d$ is the shortest distance from each link of the manipulator to the obstacles, and $D_0$ is the predefined value that defines the effective area of the potential. $\alpha_1$ and $\alpha_2$ are weights, which are decided depending on the desirability between the high manipulability and importance of obstacle-free.

3.2 Learning algorithm

At the beginning of the learning, initial parameters of neurons are set randomly so the output of joint angles $\theta^\text{out}$ will lead the end effector to a wrong position and make the manipulator take a wrong pose. In the following learning process, the parameters are amended. After many times iteration, the appropriate relation between the image spaces and the joint angle spaces will be established.

For $N$-th iteration, the maps learn with a target $u_t$ following the learning flow illustrated in Fig. 4.

(1) A target position $u_t$ is given arbitrarily in the workspace. The target positions in the images are obtained by the cameras, and then they are transferred to the self-organizing maps.

(2) The maps sort the neurons in the ascending order of distance between the target and $W_i$ of the neurons.

$$||u_t - W_1|| < \cdots < ||u_t - W_{\lambda^\text{p}}||$$

Where, $\lambda^\text{p}$ is the number of the neurons updated in the $N$-th iteration.
(3) One map outputs the joint angles $\theta^\text{out}$ using Eq. (7), and the manipulator changes its pose using $\theta^\text{out}$.

$$\theta^\text{out} = \frac{\sum_{i=1}^{p} g_o(\text{order}_{i}, \lambda^o)(\theta^i + J^i \text{out}(u^i - W^i))}{\sum_{i=1}^{p} g_o(\text{order}_{i}, \lambda^o)} \quad (7)$$

Since the right neuron parameters have not been obtained, the end effector does not reach the target. The cameras obtain the position of the end effector $v_i$.

(4) In order to reduce the error, the map changes $\theta^\text{out}$ by Eq. (8).

$$\theta^\text{out} + \frac{\sum_{i=1}^{p} g_o(\text{order}_{i}, \lambda^o)(\theta^i + J^i \text{out}(u^i - v_i))}{\sum_{i=1}^{p} g_o(\text{order}_{i}, \lambda^o)} \quad (8)$$

The cameras also obtain the position of the end effector $v_i$.

(5) The self-organizing maps correct the parameters of each neuron by the method mentioned in the following sub-sections.

### 3.3 Update of the parameters

Each parameter of the neurons are updated using $\theta^\text{out}$, $u_i$, $v_0$ and $v_1$.

(a) Updating $W$

$$W^\text{out} = W^\text{out} + e^o g_o(\text{order}_{i}, \lambda^o)(u^i - W^i) \quad (9)$$

Where, $e^o$ is the learning coefficient of $W$. It has a large value for early stages of learning process and has a small value for late stages. Updating $W$ by Eq. (9), the neurons will be distributed all over the image spaces.

(b) Updating $J$

$$J^i = J^i + e^j g_j(\text{order}_{i}, \lambda^j)\Delta J^i \quad (10)$$

Where, $e^j$ is the learning coefficient of $J$, and it changes just like $e^o$. $\Delta J^i$ is determined by Widrow-Hoff’s learning rule. The evaluation function $E_j$ is

$$E_j = \frac{1}{2} \|(v_1 - v_0) - J^i (\theta^\text{out} - \theta^\text{out})\|^2 \quad (11)$$

Then, $\Delta J^i$ is determined by

$$\Delta J^i = -C_j \frac{\partial E_j}{\partial J^i} = \frac{(v_1 - v_0) - J^i \theta^\text{out} T}{\|\theta^\text{out}\|^2} \quad (12)$$

Where, $\delta_0 = v_1 - v_0$, $\delta^\text{out} = \delta_1 - \delta_0$, $C_j = 1/\|\theta^\text{out}\|^2$. By updating $J$ using Eq. (10), the maps become to output more appropriate $\theta^\text{out}$.

(c) Updating $\xi$

$\xi$ is the gradient vector of the function $H$. $\xi$ is defined as $\frac{\partial H}{\partial \theta}$. It is obtained as the following equation.

$$\xi = \alpha_1 \xi_M + \alpha_2 \xi_O \quad (13)$$

Where, $\xi_M$ and $\xi_O$ are the gradient vectors of the function $H_M$ and $H_O$. $\alpha_1$ and $\alpha_2$ are weights.

(1) Updating $\xi_M$

$\xi_M$ is updated by Eq. (14).

$$\xi_{M_{ij}}^{n+1} = \xi_{M_{ij}}^n + e^\xi_M g_j(\text{order}_{i}, \lambda^j)\Delta \xi_{M_{ij}} \quad (14)$$

Where, $e^\xi_M$ is the learning coefficient of $\xi_M$. $\Delta \xi_{M_{ij}}$ is updated likely as $J^i \theta^\text{out}$.

The evaluation function is

$$E_\xi = \frac{1}{2} \|(H_{M_{ij}} - H_{M_i}) - \xi_{M_{ij}} \| \cdot \|J^i \theta^\text{out}\|^2 \quad (15)$$

Where, $j$ is the ID number of the neuron that has the 1st order for the distance between it and the target, and $k$ is the ID number of the neuron that has 2nd order.

Therefore, $\Delta \xi_{M_{ij}}$ becomes

$$\Delta \xi_{M_{ij}} = -C_\xi \frac{\partial E_\xi}{\partial \xi_{M_{ij}}} = \frac{(H_{M_{ij}} - H_{M_i}) - \xi_{M_{ij}} \|J^i \theta^\text{out}\|^2}{\|\theta^\text{out}\|^2} \quad (16)$$

Where, $H_{M_{ij}} = H_{M_{ij}} - H_{M_{i}}, \theta^i = \theta^i - \theta^0_i, C_\xi = 1/\|\theta^\text{out}\|^2$.

(2) Updating $\xi_O$

$\xi_O$ is updated by Eq. (17).

$$\xi_{O_{ij}}^{n+1} = \xi_{O_{ij}}^n + e^\xi_O g_o(\text{order}_{i}, \lambda^o)\Delta \xi_{O_{ij}} \quad (17)$$

Where, $e^\xi_O$ is the learning coefficient of $\xi_O$. $\Delta \xi_{O_{ij}}$ is updated likely as $\Delta \xi_{M_{ij}}$. The evaluation function is

$$E_{\xi} = \frac{1}{2} \|(H_{O_{ij}} - H_{O_i}) - \xi_{O_{ij}} \| \cdot \|\theta^\text{out}\|^2 \quad (18)$$

Where, $H_{O_{ij}}$ and $H_{O_i}$ are the potential values for $v_0$ and $v_1$.

Therefore, $\Delta \xi_{O_{ij}}$ becomes

$$\Delta \xi_{O_{ij}} = -C_{\xi} \frac{\partial E_{\xi}}{\partial \xi_{O_{ij}}} = \frac{(H_{O_{ij}} - H_{O_i}) - \xi_{O_{ij}} \|\theta^\text{out}\|^2}{\|\theta^\text{out}\|^2} \quad (19)$$

Where, $H_{O_{ij}} = H_{O_{ij}} - H_{O_i}, \theta^i = \theta^i - \theta^0_i, C_{\xi} = 1/\|\theta^\text{out}\|^2$.

By updating $\xi_M$ and $\xi_O$ using Eqs. (14) and (17), the maps become to output $\theta^\text{out}$ that makes the manipulator achieve high manipulability and obstacle-free poses.

(d) Updating $\theta$

$\theta$ is updated by Eq. (20).

$$\theta_i^{n+1} = \theta_i^n + e^\theta g_j(\text{order}_{i}, \lambda^j)\Delta \theta_i \quad (20)$$

Where, $e^\theta$ is the learning coefficient of $\theta$. $\Delta \theta_i$ becomes as

$$\Delta \theta_i = \theta_i^{\text{desire}} - \theta_i^n \quad (21)$$

$$\theta_i^{\text{desire}} = J^i \theta^\text{out} + (I - J^i \theta^\text{out} T) \xi_M \|K_p \| \quad (22)$$

Where, $K_p$ is a positive coefficient that meets the condition where high manipulability is achieved and the manipulator takes an obstacle-free pose. It decreases with learning times.

Therefore, $\Delta \theta_i$ becomes

$$\Delta \theta_i = \theta_i^n - \theta_i^{n+1} + J^i \theta^\text{out} + (I - J^i \theta^\text{out} T) \xi_M \|K_p \| \quad (23)$$

By updating $\theta$ using Eq. (20), the maps become to output $\theta^\text{out}$ for the manipulator with less error of the end effector, high manipulability and obstacle-free poses.
4. Simulation Results and Experimental Scheme

4.1 Simulation results

To evaluate the effectiveness of the proposed method, we have conducted simulation experiments. There are a 4-DOF manipulator, three CCD cameras as well as an obstacle as shown in Fig. 1 in the virtual environment. The self-organizing maps were trained with 15,000 targets given in the workspace and each map consisted of 240 neurons. Each camera has $512 \times 512$ [pixel] resolution.

After learning, 80 targets were given randomly to test the positioning accuracy of the end effector and to confirm the poses of the manipulator. In view of the visibility of those targets, the map1 or map2 output the joint angles of the manipulator. The learned maps output the joint angles of the manipulator for each target, and then the manipulator was driven. Figures 5–8 show the positions of the end effector and poses of the manipulator seen in each camera. Figures 5 and 6 show those cases where targets are visible in two maps, both map1 and map2 output joint angles for them. Figure 6 shows the case that the projections of the targets cross the projection of the obstacle in the all front cameras’ images, and then collision avoidance is confirmed in the camera3 image space. Figures 7 and 8 show those cases where targets are occluded by the obsta-
Fig. 9 Output for manipulator using map2 with an L-shape obstacle

Fig. 10 Output for manipulator using map1 with an L-shape obstacle

Fig. 11 Output for manipulator using map1 and map2 with a step-like obstacle

cle or overlapped in the front camera images (i.e. camera 1 or 2). In these cases only map1 or map2 is available to output the joint angles of the manipulator for them. In previous studies, these targets were treated as inaccessible parts for the manipulator. In the presented figures, we can see that the manipulator takes obstacle-free poses and the end effector reaches the targets precisely. The average positioning error of the end effector is 0.22 pixels.

In above simulations, the visuo-motor system has shown a good performance with a rectangular obstacle. Here, we intend to demonstrate the effectiveness of the proposed self-organizing procedures with complex-shaped obstacles. The more complicate shapes obstacles hold, the more occlusion happens. Thereby the cooperative and complementarity of multiple maps are more superior. Figures 9 and 10 show us the performance of the system with an L-shape obstacle. In Fig. 9, those targets distributed in the corner of the obstacle are invisible for the map1, so only map2 is available. Figure 10 shows the case that the targets are invisible for the map2 but visible for the map1. In addition, Figs. 11–13 show another simulation result with a step-like obstacle. Figure 11 shows the targets which are visible in two maps, and both map1 and map2 output joint angles for them. Figures 12 and 13 show the targets which are occluded by the obstacle or overlap the obstacle in one front camera’s image. The complementarity of two maps make it is possible to control the manipulator reaching to the targets. From these figures, we see that the maps are also ability to output correct joint angles $\theta_{\text{out}}$ for those targets, which are distributed along the shape of the obstacle. The simulations showed the coordination performance of the system with respect to obstacles with different outline. It is confirmed
that this learning procedure can be used with various obstacles. And this approach opens perspectives to be used with more complicated obstacles by using a more flexible manipulator with higher degree of freedom.

The simulation results presented below are to demonstrate robustness of our visuo-motor system against geometry changes of the system. While there happened any change in its geometry for a learned visuo-motor system, the system does not need to learn again from a complete unknown state. We use the learned self-organizing map as an initial map for the changed system. With such an initial map, a new appropriate self-organizing map will be reconstructed corresponding to the changed system in small learning times. Such an ability of the self-organizing map, which copes with changes immediately, is called adaptability here. Some tests of adaptability that involve changes of the manipulator position, obstacles position, and cameras position have been done. Here, we intend to show the case of changing the position of the cameras. Figures 14 and 15 show the errors of the end effector and the poses of the manipulator using un-relearned maps with respect to environmental changes. The average positioning error of the end effector is up to 16.34 pixels. However, the system is capable of requiring the appropriate relations in the changed environment by means of re-learning in limited times. Compared to Figs. 14 and 15, Figs. 16 and 17 show the results by using the maps re-learned for only 1 000 times. The average errors are decreased to 3.17 pixels.

4.2 Experimental system

We have proved the effectiveness of our approach by
the simulations in the preceding subsection. In this sub-
section, we present an experimental system, which is ap-
plied to a visuo-motor system with a 4-DOF manipulator
and 3 CCD cameras in a real environment, to validate the
proposed approach.

An outline of our experimental system is shown in
Fig. 18. While a target $u_t$ is given in the workspace, the
self-organizing maps, which are composed with a host
computer, output the joint angles of the manipulator $\theta^{out}$. The manipulator is driven by using the operating signal $\theta^{out}$ that is received through a Drive Unit. To keep the problem of image segmentation simple during the exper-
iment, LEDs (light emission diode) are mounted on the
joints so that the system can discriminate the positions of
the end effector and the links of the manipulator against
backgrounds easily. The CCD cameras see the manip-
ulator, and the outputs from the cameras are lead to a
frame memory in the host computer. The size of the frame
is $512 \times 512$ [pixel]. The CCD cameras are also used to
get shapes of obstacles. An Image Processing Board pro-
cesses the visual information obtained by the cameras, so
the positions of the end effector and the poses of the ma-
nipulator in the image spaces can be obtained. All infor-
mation is passed to the host computer. After the learning
procedure is iterated for defined times, the correct self-
organizing maps will be constructed.

Because it takes much time for the system to move
the manipulator and process the images in the experiment,
it leads to take much time to finish the learning of the self-
organizing maps as well. According to our previous re-
searches in Ref. (13), the total time required to complete 1,000 times learning was about 7 hours with the limiting factors being the manipulator moving speed and the LEDs lighting. In order to reduce the learning expenditure in a real experiment, we utilize the adaptability of the self-organizing maps that has been proved in the above subsection. First, the CCD cameras obtain the images of the manipulator and obstacles in the real environment, and then the system constructs the self-organizing maps by simulation. The time required to complete the learning was about 5 minutes (A PC with 3.0 GHz Pentium 4 was used). Then, the result of the simulation is taken as the initial values in the experiment, and the system reconstructs the self-organizing maps by the real devices only for small times. Therefore, the learning time in real experiment can be decreased obviously.

5. Conclusion

Following the previous researches, an improved method has been proposed which uses multiple self-organizing maps and a 3-camera system to achieve coordination of an uncalibrated visuo-motor system and to avoid the occlusion problem. This approach is different from other existing methods in (1) By employing two maps alternately, the system overcomes the occlusion problems in cluttered environments. The cooperative and complementarity of two maps make the manipulator move in whole workspace consistently. (2) In our self-organizing learning procedure, the visuo-motor system are learned not only to position the end effector precisely but also to ensure the manipulator acquiring obstacle-free poses. Simulation results demonstrated the efficiency of the proposed method and the robustness of the system. Validating the proposed approach by a real experiment is our next work. In addition, in the present system whether the path of the manipulator collides with obstacles or not has not been considered. Future work, however, will have to address an efficient path planning for the manipulator to realize collision avoidance.

References