Developmental Approach to Spatial Perception for Imitation Learning: Incremental Demonstrator's View Recovery by Modular Neural Network

Yuichiro Yoshikawa, Minoru Asada, and Koh Hosoda

Dept. of Adaptive Machine Systems
Graduate School of Engineering Osaka University
Suita, Osaka 565-0871, Japan
e-mail: yoshikawa@er.ams.eng.osaka-u.ac.jp, asada@ams.eng.osaka-u.ac.jp
hosoda@ams.eng.osaka-u.ac.jp

Abstract

Imitation Learning is not simply one of the most promising ways to accelerate the behavior acquisition for humanoid robots but also one of the most interesting cognitive issues to model how we human beings learn to acquire various kinds of behaviors. As the first step towards developmental approach to spatial perception for imitation learning, this paper proposes a method of incremental recovery of the demonstrator's view using a modular neural network by which the learner can organize spatial perception for the view-based imitation learning with the demonstrator in different positions and orientations.

1 Introduction

Imitation learning is one of the most promising ways to accelerate the behavior acquisition for many DOF robots [1]. Because, machine learning theories seems difficult to directory apply to real robot tasks as they are due to the huge search space caused by multi-modal sensor space and many DOFs which also add much more uncertainties than computer simulations. Another aspect of the research on imitation learning is that it is also one of the most interesting cognitive issues to model how we human beings learn to acquire various kinds of behaviors by building real robots capable of imitation learning [2].

Since existing robotic approaches have focused on behavior generation and memorization, they assumed the powerful observation capabilities. These robots can estimate the internal state of the demonstrator given the 3-D geometrical parameters (ex. [3, 4, 5]) or the coordinate transformation from the demonstrator's to the learner's (ex. [6, 7]). However, such assumptions do not seem applicable to the case of imitation learning by human. Therefore, the robot should learn behaviors by using its on-board sensors from a viewpoint of cognitive approaches.

Asada et al. proposed the method of the view-based approach to imitation learning without assuming the observation of the demonstrator's internal state [8]. In their approach, the learner can recover the demonstrator's view based on opt-geometric constraint (stereo epipolar geometry [9]), assuming that the demonstrator has the same body structure as the learner's. By realizing the same trajectories of demonstration in the recovered view, the learner can estimate the internal state of demonstrator.

Since the parameters of their system depend on its viewpoint (body orientation of the demonstrator to itself), the learner needs to recover the demonstrator's view from the beginning if the body orientation of the demonstrator changes. From a viewpoint of the cognitive approach, the learner is expected to cope with the change of the body orientation of the demonstrator. In other words, it is a very interesting issue how the learner develops its spatial perception for imitation from the visio-motor map learning, and the capability of internal state estimation for the demonstrator. Then, we propose a method of incremental recovery of the demonstrator's view using modular neural network by which the learner can organize spatial perception for the view-based imitation learning with the demonstrator in different positions and orientations.

The rest of the paper is organized as follow: first, the previous work of demonstrator's view recovery based on epipoalr geometry is revisited partially because of explaining the background of the problem and partially because of showing the existence of the solu-

Proceedings of the IEEE-RAS Internatinonal Conference on Humanoid Robots

tion. Next, the proposed method is given with a modular network architecture. Then, the experimented results are shown and finally discussion and future work are given.

2 Previous work [8] : Demonstrator's view recovery based on epipolar geometry

2.1 Epipolar geometry

Fig.1 shows an epipolar constraint between a pair of stereo images [p] and [q]. Given a point ${}^{p}\boldsymbol{m}_i$ (${}^{q}\boldsymbol{m}_i$) in the left (right) image [p] ([q]), its corresponding point ${}^{q}\boldsymbol{m}_i$ (${}^{p}\boldsymbol{m}_i$) in the right (left) image[q] ([p]) is constrained to lie on a line called *epipolar line*. This relationship (constraint) between two cameras is called *epipolar geometry*.

If these two cameras can be approximated by affine camera model [10], epipolar geometry is given by

$$ {}^{pq}\boldsymbol{u}_i^T {}^{pq}\boldsymbol{f} + {}^{pq}f_{33} = 0, \tag{1} $$

where ${}^{pq}\boldsymbol{u}_i = [{}^{p}m_i, {}^{q}m_i]^T$ is a vector consists of the i-th matched image point between [p] and [q], and ${}^{pq}\boldsymbol{f} = [{}^{pq}f_{13}, {}^{pq}f_{23}, {}^{pq}f_{31}, {}^{pq}f_{32}]^T$ consists of nonzero elements in the affine fundamental matrix for the epipolar geometry between [p] and [q].

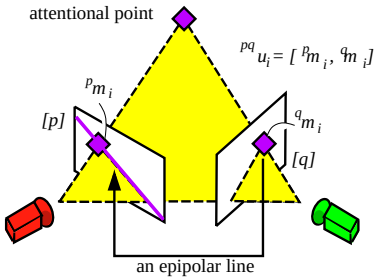


Figure 1: epipolar geometry

2.2 Estimation of affine fundamental matrix [10]

In general, a minimum of 4 pairs of matched points are required to uniquely determine the affine fundamental matrix. It can be determined by minimizing the sum of distances of each point ${}^{pq}\boldsymbol{u}_i$ to the hyperplane (eq.1) in the 4-dimensional space.

By this method, the affine fundamental matrix ${}^{pq}\boldsymbol{f}$ is determined as an eigenvector associated with the minimal eigenvalue of ${}^{pq}\boldsymbol{W}$, where

$$ {}^{pq}\boldsymbol{W} = \sum_{i=1}^{N} ({}^{pq}\boldsymbol{u}_i - \frac{1}{n}\sum_{j=1}^{N} {}^{pq}\boldsymbol{u}_j)({}^{pq}\boldsymbol{u}_i - \frac{1}{n}\sum_{j=1}^{N} {}^{pq}\boldsymbol{u}_j)^T, \tag{2} $$

then,

$$ {}^{pq}f_{33} = -{}^{pq}\boldsymbol{u}_0^T {}^{pq}\boldsymbol{f}. \tag{3} $$

2.3 Finding a corresponding point in a different view

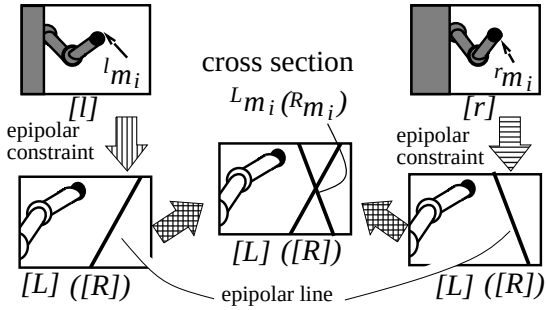


Figure 2: Overview of the method to find a corresponding point in an added view [L] ([R]) from the two views [l] and [r].

We add one more camera [L] ([R]) observing a point which is also observed in [l] and [r]. The problem is how to find the corresponding points in the view [L] ([R]) with ones in the views [l] and [r].

Based on epipolar constraints, the matched points ${}^{l}\boldsymbol{m}_i$ (${}^{r}\boldsymbol{m}_i$) are constrained to lie on the epipolar lines on [L] ([R]). We can find the matched points ${}^{L}\boldsymbol{m}_i$ (${}^{R}\boldsymbol{m}_i$) on the cross sections of epipolar lines (see Fig.2).

Thus, there are four epipolar equations ([l,L], [r,L], [l,R], and [r,R]) to be satisfied. Expanding their formulations algebraically, we obtain

$$ \boldsymbol{A}\,{}^{LR}\boldsymbol{u}_i = -\boldsymbol{B}\boldsymbol{c}_i, \tag{4} $$

where

$$ \boldsymbol{A} = \begin{bmatrix} {}^{lL}f_{31} & {}^{lL}f_{32} & 0 & 0 \\ {}^{rL}f_{31} & {}^{rL}f_{32} & 0 & 0 \\ 0 & 0 & {}^{lR}f_{31} & {}^{lR}f_{32} \\ 0 & 0 & {}^{rR}f_{31} & {}^{rR}f_{32} \end{bmatrix} $$

$$ \boldsymbol{B} = \begin{bmatrix} {}^{lL}f_{13} & {}^{lL}f_{23} & {}^{lL}f_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^{rL}f_{13} & {}^{rL}f_{23} & {}^{rL}f_{33} \\ {}^{lR}f_{13} & {}^{lR}f_{23} & {}^{lR}f_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^{rR}f_{13} & {}^{rR}f_{23} & {}^{rR}f_{33} \end{bmatrix} $$

$$ \boldsymbol{c}_i = [{}^{l}x_i \quad {}^{l}y_i \quad 1 \quad {}^{r}x_i \quad {}^{r}y_i \quad 1]^T $$

If $\boldsymbol{A}^T\boldsymbol{A}$ is nonsingular, ${}^{LR}\boldsymbol{u}_i$ is determined from the matched points of the other two stereo images, such

Proceedings of the IEEE-RAS Internatinonal Conference on Humanoid Robots Copyright ©2001

as,

$$ {}^{LR}\boldsymbol{u}_i = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{c}_i. \quad (5) $$

Following this equation, projected points on [l] and [r] are transformed onto [L] and [R]. Hereafter we call this mechanism as view transformation mechanism, which finds corresponding points on a certain stereo images to projected points on different ones.

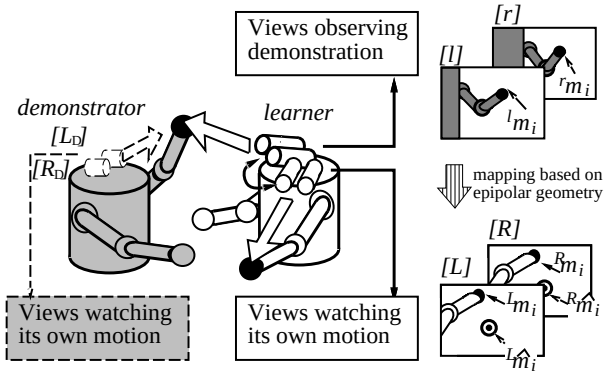


Figure 3: The learner's view when observing ([l], [r]) and imitating ([L], [R]), and the demonstrator's view ([L_D], [R_D]) when watching itself.

2.4 Demonstrator's view recovery

Hereafter, we assume that the learner and the demonstrator have the same body structure, that is, the same link structure and the same camera parameters. Then, the demonstrator's stereo views ([L_D] and [R_D]) watching its self motion can be regarded as the learner's ones ([L] and [R]) watching its self motion if the learner succeed in exactly imitating the demonstrator's motion. If we can recover the demonstrator's view, it means the learner's view to be realized. Then, the problem is how to recover the views ([L_D] and [R_D]) as the views ([L] and [R]) based on epipolar geometry (see Fig.3).

We can determine all affine fundamental matrices between [l],[r],[L] and [R] by assuming that initial postures of both the demonstrator and the learner are the same and the corresponding points on the body parts are given. Using these affine matrices, we can recover the demonstrator's view which shows the desired trajectories for the learner to realize based on the method described in 2.3. Then, we can apply adaptive visual servoing method to imitate the demonstrator's motion [8].

Proceedings of the IEEE-RAS Internatinonal Conference on Humanoid Robots Copyright ©2001

3 View transformation mechanism by modular architecture

In the previous work [8], the learner must re-estimate the parameters of view transformation mechanism everytime its viewpoint to the demonstrator changes. In order to cope with this problem, we propose a view transformation mechanism by modular architecture. It consists of a number of modules of view transformation and integrates the outputs from these modules.

The merits of modular architecture are:

- by assigning different module for the view transformation in different learner's viewpoint, it can store view transformation mechanisms, and
- by integrating a number of modules cooperatively, it can realize the view transformation for unexperienced viewpoints.

Although there are many possible modular structures for view transformation mechanism, we apply the liner network here for simplicity (see Fig.4). In order to realize the modular learning system, we must define

- how each module computes its output and learns to converge it to the desired value,
- how to integrate the outputs of the modules, and
- how to assign the responsibilities of learning to each module.

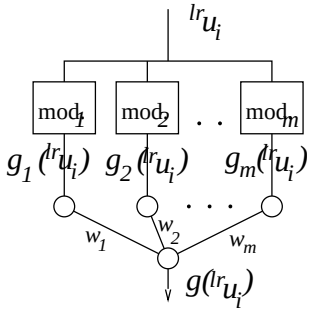


Figure 4: The modular architecture which integrates the outputs of modules by the liner sum.

3.1 The structure of the individual module

It is a straightforward way to apply the view transformation mechanism based on epipolar geometry (eq. (5)) to the structure of the module in the modular architecture. However, in the consideration of the compatibility with the following algorithm of the incremental learning, we substitute matrix operation in eq.

(5) with a neural network. In order to let it converge to the desired value based on epipolar geometry, the structure should reflect the formulation described in 2.3.

Thus, the matrix operation in eq. (5) can be substituted with almost full connection feedforward neural network in which the unit function is liner and some connection are cut since the parameter matrix B in equation (5) contains zero elements (see Fig.5). Instead of the top-down architecture with the full knowledge of the matrix B, we use a sigmoidal function by which we expect the generalization capability.

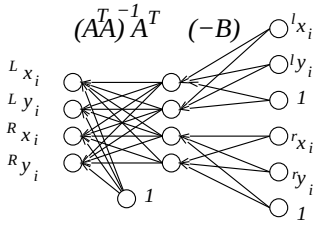


Figure 5: The structure of view transformation module

3.2 Integration of the outputs in the modular architecture

Suppose that the learner observes m feature points on the demonstrator's body in the views ($[l]$ and $[r]$), and it also observes m corresponding feature points on the learner's one in the views ($[L]$ and $[R]$). Let the feature vector of the i-th feature point in the views $[l]$ and $[r]$ be ${}^{lr}\boldsymbol{u}_i = [{}^l\boldsymbol{m}_i^T, {}^r\boldsymbol{m}_i^T]^T$, where ${}^v\boldsymbol{m}_i = [{}^vx_i, {}^vy_i]^T$ is an image coordinates of the i-th feature point in the view $[v]$. Further, let the feature vector of the i-th feature point in the views $[L]$ and $[R]$ be ${}^{LR}\boldsymbol{u}_i = [{}^L\boldsymbol{m}_i^T, {}^R\boldsymbol{m}_i^T]^T$.

In our modular structure, each module receives the feature vector as an input and then outputs a vector. The transformation of j-th module can be described as a function $\boldsymbol{g}_j \in \Re^4$. Thus, when the i-th feature vector ${}^{lr}\boldsymbol{u}_i$ is input, the output of the j-th module ($j = 1, \cdots, m$) is $\boldsymbol{g}_j({}^{LR}\boldsymbol{u}_i)$. The network integrates the output by liner sum of each module's output. The transformation of the network can also be described as a auction $\boldsymbol{g}$. When the i-th feature vector ${}^{lr}\boldsymbol{u}_i$ is input, the network output can be described as

$$\boldsymbol{g}({}^{lr}\boldsymbol{u}_i) = \sum_{j=0}^{m} w_j \boldsymbol{g}_j({}^{lr}\boldsymbol{u}_i). \tag{6}$$

Since the purpose of this network is to realize the view transformation mechanism, all n network outputs $\boldsymbol{g}({}^{lr}\boldsymbol{u}_i), i = 1, \cdots n$ should correspond to the learner's feature vector ${}^{LR}\boldsymbol{u}_i$. Therefore, the average of the squared transforming errors about all feature points,

$$E = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{g}({}^{lr}\boldsymbol{u}_i) - {}^{LR}\boldsymbol{u}_i|^2 \tag{7}$$

should be minimized.

To minimize E, it is a possible way to let the output of better module which has less transforming error ($\epsilon_j = |\boldsymbol{g}_j({}^{lr}\boldsymbol{u}_i) - {}^{LR}\boldsymbol{u}_i|^2$) contribute to network output more. It can be realized by making the weight w_j in the eq. (6) be a soft-max function of the module's transforming error,

$$w_j = \frac{exp(\epsilon_j/\sigma^2)}{\sum_{k=0}^{n} exp(\epsilon_k/\sigma^2)}, \tag{8}$$

where σ is a scalar parameter which determines the degree of evaluation of the transforming error.

3.3 Learning in the modular architecture

Competitive learning seems a promising way to let the network learn the view transformation mechanism incrementally. It can be realized by controlling the learning rates of modules. That is, better modules assigned more learning weight learn more while worse modules less, and then, the better ones will be a expert of the view transformation mechanism in a certain situation. In a different situation, a different module would be a expert. Thus, competitive learning is realized.

Such competitive learning is implemented by using a weight in eq. (8) as a learning rate for the j-th module. Each module learns by back-propagation method in which the error vectors at the output layer is multiplied by the weight.

In order to let more modules learn when the network output is less close to the desired value, the parameter σ in eq. (8) is modulated as

$$\sigma^2 = kE. \tag{9}$$

4 Experiments

To show the validity of the proposed method, a number of experiments are performed. Two identical manipulators are assumed as bodies of the learner and the demonstrator, respectively. A pair of stereo cameras corresponds to larner's view point(see Fig.6).

In this experiment, the learner observes six markers on each manipulator as features on the each body. Using color tracking vision system (FUJITSU), the learner tracks them on the observed image (640×480 [pixel]) which includes the stereo images combined by field multiplexer. The baseline is about 60 cm.

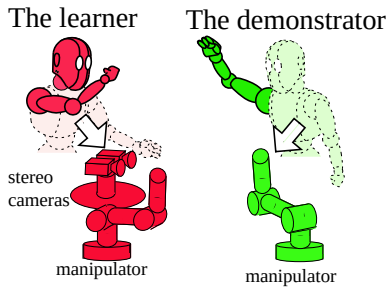
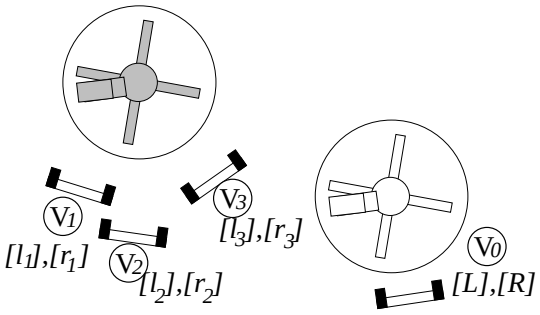


Figure 6: Assuming the learner and the demonstrator as two identical manipulators.

4.1 The capability of the neural network module

In order to confirm whether each neural network module can realize the view transformation mechanism, an experiment using only one neural network module is shown first. Given the views V_1 ($[l_1]$,$[r_1]$) observing the demonstrator and those V_0 ([L],[R]) observing the learner itself, the neural network learns the view transformation from V_1 to V_0.

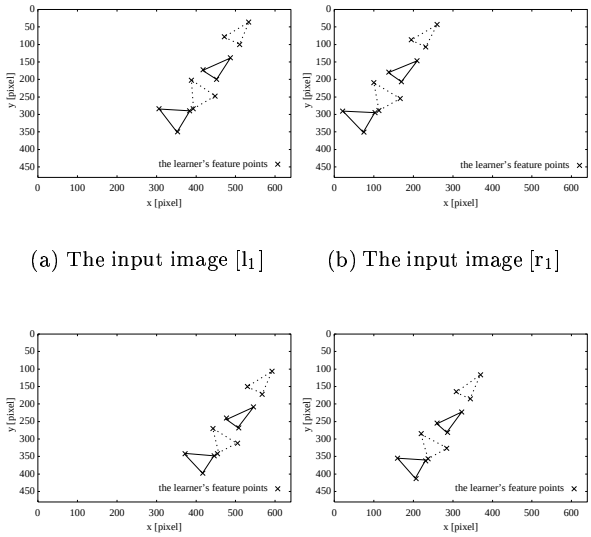


(a) The views of observing the demonstrator

(b) The view of observing the learner itself

Figure 7: The viewpoints in observing the demonstrators (a) and in observing the learner itself. V_i indicates a view point ID for the i-th view point.

Assuming that the postures of both the learner and the demonstrator are the same, we can use the feature vector ${}^{LR}\boldsymbol{u}_i$ as a teacher vector for a corresponding input feature vector ${}^{l_1r_1}\boldsymbol{u}_i$. In this experiment, instead of using many feature points, the learner observes six ones on the demonstrator and on itself at different two postures. Thus, the learner can use twelve feature points for learning. Fig.8 shows the input vector set ${}^{l_1r_1}\boldsymbol{u}_i$ (Fig.8a, b) and the desired vector set ${}^{LR}\boldsymbol{u}_i$ (Fig.8c, d).



(a) The input image $[l_1]$

(b) The input image $[r_1]$

(c) The desired image [L]

(d) The desired image [R]

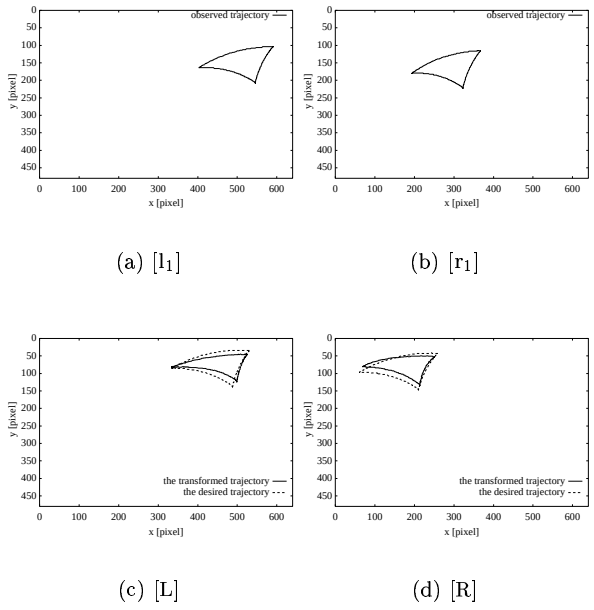
Figure 8: The input (on $[l_1]$,$[r_1]$) and desired output (on [L],[R]) feature vectors which correspond to observed features on both body parts.

The network receives the feature vectors in $[l_1]$,$[r_1]$ ${}^{l_1r_1}\boldsymbol{u}_i$, $i = 1, \cdots, 12$ which are on the demonstrator's body, and adapts itself to output the feature vectors on [L],[R] ${}^{LR}\boldsymbol{u}_i$, $i = 1, \cdots, 12$ by backpropagation method. Instead of updating the synaptic weights in every backpropagation, we update them by sum of the results of backpropagation of each feature vector.

In order to confirm that the network after learning realizes the view transformation mechanism, we input the unexperienced trajectories of demonstration on $[l_1]$,$[r_1]$ (see Fig.9 (a, b)). The transformed trajectories are shown in Fig.9 (c, d) with the true trajectories on [L],[R] which is observed when the learner generates the same motion as demonstration. Since these trajectories are almost the same, we may conclude that the neural network module has a potential of view transformation mechanism.

4.2 Incremental learning of modules

In order to show the potential of the proposed network for incremental learning, some experimental results are shown. In this setup, at first the network learns the view transformation from $[l_1]$,$[r_1]$ to [L],[R], in which a certain module becomes responsible for the transformation. Then, given new views V_2 ($[l_2]$,$[r_2]$) and V_3 ($[l_3]$,$[r_3]$), the network learns additional both



(a) [l_1] (b) [r_1]

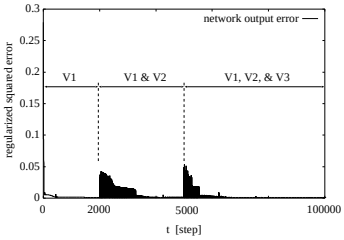
(c) [L] (d) [R]

Figure 9: The input trajectories in view [l_1] (a),[r_1] (b) to the learned neural network and the output trajectories of it with the true trajectory in view [L] (c), [R] (d).

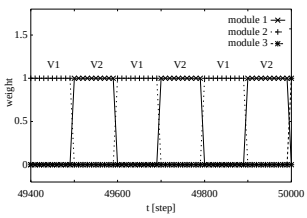
view transformations from V_2 to V_0, and from V_3 to V_0. Competitive learning, in which different modules become responsible for different view transformations, is realized in this additional learning.

In the first 100 steps, all modules learn with fixed responsibility weights (each has about 0.33). After that, the network outputs and learns based on responsible weights which are computed by eq. (8). In the first period ($0 \sim$ 20,000-th step), the network learns only one view transformation (from V_1 to V_0), and in the second (20,001 $\sim$ 50,000-th step), it learns two view transformation (from V_1 to V_0 and from V_2 to V_0). In the final step, adding one more view, it learns three transformation mechanism (from V_1 to V_0, from V_2 to V_0 and from V_3 to V_0).

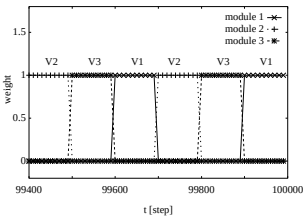
Fig.10(a) shows the learning curves and view transition to be learned during the learning process. The transition of responsible weight is shown both in learning two views (Fig.10b) and in learning three views (Fig.10c). In these figures, the label V_i indicates which view transformation is to be learned. Since each weight of a module becomes 1 exclusively, we can understand each module becomes responsible for its own view transformation, such as (1st module, from V_2 to V_0), (2nd module, from V_1 to V_0), and (3rd module, from V_3 to V_0). Thus, competitive learning is realized.



(a) Learning curve



(b) Transition of responsible weight in learning two transformation mechanism



(c) Transition of responsible weight in learning there transformation mechanism

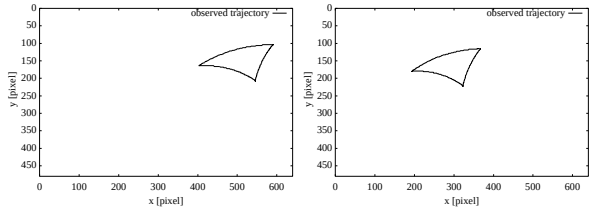
Figure 10: The learning curve and transition of each weight of a module in the incremental learning task.

In order to confirm whether the network after learning realizes the view transformation mechanism, we input the unexperienced trajectories of demonstration in the views [l_1],[r_1] (see Fig.11 (a, b)), [l_2],[r_2] (see Fig.12 (a, b)), and [l_3],[r_3] (see Fig.13 (a, b)) which are not included in learning data set. The transformed trajectories by the proposed method (MNN) are shown in Fig.11 (c, d), Fig.12 (c, d) and Fig.13 (c, d) with the true ones (true) on [L],[R] which is observed when the learner generates the same motion as demonstration, and also with the transformed ones (matrix) by the matrix operation in the eq. (5). Since the trajectories by the proposed method are are close to true one, we may conclude that the modular network has a potential of incremental learning of view transformation mechanism. In Fig.13, the trajectories by the matrix operation are very noisy. From this, it can be said that the matrix operation is more sensitive than the proposed method.

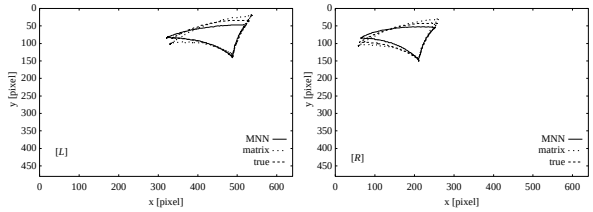
5 Discussion

The proposed network for view transformation determines contribution of each module for output de-

Proceedings of the IEEE-RAS Internatinonal Conference on Humanoid Robots Copyright ©2001

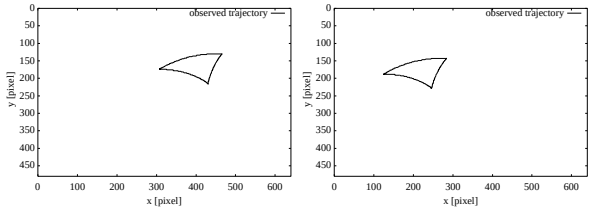


(a) [l_1] (b) [r_1]

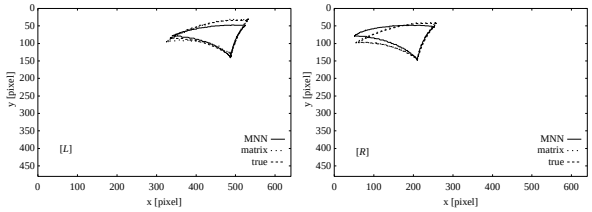


(c) [L] (d) [R]

Figure 11: The input trajectories on view [l_1] (a),[r_1] (b) and the output trajectories of the proposed method (MNN) and the matrix operation (matrix) with the true ones (true) on view [L] (c), [R] (d).



(a) [l_2] (b) [r_2]



(c) [L] (d) [R]

Figure 12: The input trajectories on view [l_2] (a),[r_2] (b) and the output trajectories of the proposed method (MNN) and the matrix operation (matrix) with the true ones (true) on view [L] (c), [R] (d).

pending on there transforming errors. In order to evaluate them, it needs some reference between views. For this reason, we assume that the posture of the learner and the demonstrator is the same. Since this assumption seems unrealistic, we need to cope with a problem how the learner knows the view transformation when the both postures are different each other.

In order to cope with the problem, Yoshikawa et al. [11] proposed the method which simultaneously estimates a corresponding posture with the demonstrator and view transformation by the control to minimize the transformation error. Although this method is for the view transformation mechanism based on epipolar geometry, we tried to apply this idea to the proposed modular neural network.

In our preliminary experiment, the learner can find the corresponding posture and correct view transformation only when the learner's initial posture is close to the demonstrator's. However it may imply that the problem can be solved if the learner can use some references. Considering the situation of human imitation, it seems to be natural that the learner can use some feature points as references, for example chest, shoulder, hip and so on.

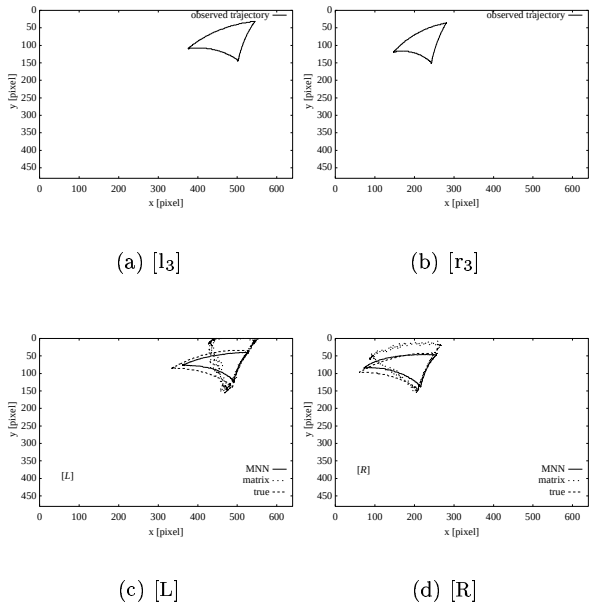
Currently, we have not used the knowledge such as rigidity between feature points although we assume the body structure of the learner and the demonstrator is the same. Such knowledge may help in searching the reference between the learner and the demonstrator. This is our future work.

Neither, we have applied the method to output unlearned view transformation, which might be represented by a method to cooperatively integrate modules. The problem is how to develop such a method, which it is our future work.

6 Conclusion

In this paper, we proposed an architecture for view transformation mechanism in the context of view-based imitation. It has modules which have neural network structure reflecting the view transformation mechanism based on epipolar geometry, and learn view transformation incrementally. These modules are integrated and learn based on closeness between the desired value and outputs. In the real robot experiment, we demonstrated that the proposed architecture learn view transformation incrementally.

Proceedings of the IEEE-RAS Internatinonal Conference on Humanoid Robots Copyright ©2001



(a) [l_3] (b) [r_3]

(c) [L] (d) [R]

Figure 13: The input trajectories on view [l_3] (a),[r_3] (b) and the output trajectories of the proposed method (MNN) and the matrix operation (matrix) with the true ones (true) on view [L] (c), [R] (d).

References

[1] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Science*, Vol. 3, No. 6, pp. 233–242, 1999.

[2] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi. Cognitive developmental robotics as a new paradigm for the design of humanoid robots. In *Proc. of the 1st IEEE-RSA International Conference on Humanoid Robots*, 2000.

[3] T. Inamura, I. Toshima, H. Ezaki, and Y. Nakamura. Generation of whole body motion using mimesis loop and primitive symbols(in japanese). In *Proc. of the 18th Annual Conference of the Robotics Society of Japan*, pp. 801–802, 2000.

[4] A. Billard and Maja Mataric. Learning human arm movement by imitation: Evaluation of a biologically-ispired connectionist architecture. In *Proc. of the 1st IEEE-RAS International Conference on Humanoid Robots*, 2000.

[5] K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation. *IEEE Trans. on R & A*, pp. 368–385, 1994.

[6] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato. A kendama learning robot based on bi-directional theory. *Neural Networks*, Vol. 9, pp. 1281–1302, 1996.

[7] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Trans. on R&A*, Vol. 10, No. 6, pp. 799–821, 1994.

[8] M. Asada, Y. Yoshikawa, and K. Hosoda. Learning by observation without three-dimensional reconstruction. In *Intelligent Autonomous Systems*, pp. 555–560, 2000.

[9] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene of from two projections. *Nature*, Vol. 293, pp. 133–135, 1981.

[10] G. Xu and Z.Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach.* Kluwer Academic Publisher, 1996.

[11] Y. Yoshikawa, M. Asada, and K. Hosoda. View-based imitation learning by conflict resolution with epipolar geometry (to appear). In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.