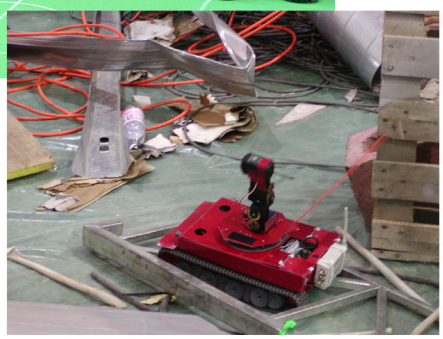
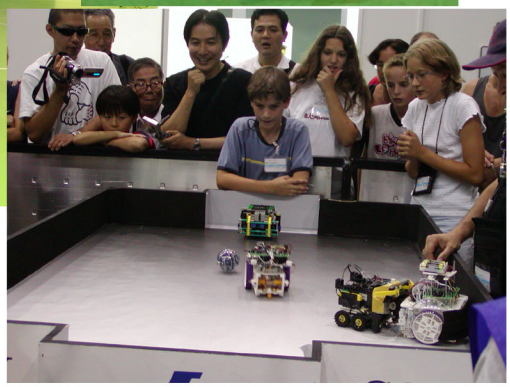
人工知能学会
第19回SIG-Challenge研究会





2004

RoboCup

2004年5月2日
インテックス大阪
（ロボカップジャパンオープン2004）

目次

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-0419-1 (5/1)

Adaptive Behavior Acquisition based on Modular Learning System and Scheduling in Multi-Agent Environment

Yasutake Takahashi, Kazuhiro Edazawa, and Minoru Asada
Dept. of Adaptive Machine Systems, Graduate School of Engineering, Osaka University, Osaka, Japan
{yasutake,eda,asada}@er.ams.eng.osaka-u.ac.jp

Abstract

The existing reinforcement learning approaches have been suffering from the policy alternation of others in multiagent dynamic environments such as RoboCup competitions since other agent behaviors may cause sudden changes in state transition probabilities of which constancy is needed for the learning to converge. A modular learning approach would be able to solve this problem if we can assign each module to one situation in which the module can regard the state transition probabilities as constant. This paper presents a method of modular learning in a multiagent environment, by which the learning agent can adapt its behaviors to the situations as consequences of the other agent's behaviors. Scheduling for learning is introduced to avoid the complexity in autonomous situation assignment.

1 Introduction

There have been an increasing number of approaches to robot behavior acquisition based on reinforcement learning methods [Asada et al., 1996, Connell and Mahadevan, 1993]. The conventional approaches need an assumption that the environment is almost stationary or changing slowly so that the learning agent can regard the state transition probabilities as constant during its learning. Therefore, it seems difficult to directly apply the reinforcement learning method to a multiagent system because a policy alteration of other agents may occur, which dynamically changes the state transition probabilities from the viewpoint of the learning agent. RoboCup provides such a typical situation, that is, a highly dynamic, hostile environment, in which agents have to obtain purposive behaviors.

There are a number of studies on reinforcement learning systems in a multiagent environment. Asada et al. [Asada et al., 1999] proposed a method which estimates the state vectors representing the relationship between the learner's behavior and those of other agents in the environment using a technique from system identification, then reinforcement learning based on the estimated state vectors is applied to obtain a cooperative behavior. However, this method requires a global learning schedule in which only one agent is specified as a learner and the rest of agents have a fixed policies. Therefore, the method cannot handle the alternation of the opponents policies. This problem happens because one learning module can maintain only one policy. A modular learning approach would provide one solution to this problem. If we can assign multiple learning modules to different situations in each of which module can regard the state transition probabilities as constant, then the system could show a reasonable performance.

Jacobs and Jordan [Jacobs et al., 1991] proposed the mixture of experts, in which a set of the expert modules learn and the gating system weights the output of the each expert module for the final system output. This idea is very general and has a wide range of applications. Singh [Singh, 1992c, Singh, 1992a] has proposed compositional Q-learning in which an agent learns multiple sequential decision tasks with multi learning modules. Each module learns its own elemental task while the system has a gating module which learns to select one of the elemental task modules. However, there are no such measure to identify the situation that the agent can switch modules corresponding to the change of the situation. Tani and Nolfi [Tani and Nolfi, 1997a, Tani and Nolfi, 1997b] extended the idea to mixture of recurrent neural network and introduced it to predict sensory flow pattern under the robot navigation task. Their scheme, however, doesn't have any control learning structure, which makes it difficult to acquire a purposive behavior by itself. Doya et al. [Doya et al., 2000] have proposed MOdular Selection and Identification for Control (MOSAIC), which is a modular reinforcement learning architecture for non-linear, non-stationary control tasks. Their idea was applied to relatively simple tasks/dynamic environment, however, it is uncertain that it is possible to assign modules automatically in the multi-agent system that has highly dynamic ones.

We adopt the basic idea of the mixture of experts into an architecture of behavior acquisition in the multi-agent environment. In this paper, we propose a method by which multiple modules are assigned to different situations and learn purposive behaviors for the specified situations which are expected as the consequence of other agent's behavior under different policies. Takahashi et al. [Takahashi et al., 2002] have shown preliminary experimental results under same domain, however, the learning modules were assigned by the human designer. In this paper, a scheduling for learning is introduced to avoid the complexity in autonomous situation assignment.

2 A Basic Idea and An Assumption

The basic idea is that the learning agent could assign one behavior learning module to each situation which is caused by the other agents and the learning module would acquire a purposive behavior under the situation if the agent can distinguish a number of situations in which the state transition probabilities are almost constant. We introduce a modular learning approach to realize this idea. A module consists of learning component that models the world and an execution-time planning one. The whole system performs the following procedures simultaneously.

- find a model which represents the best estimation among the modules,
- update the model, and
- calculate action values to accomplish a given task based on dynamic programming (DP).

As an experimental task, we prepare a case of ball passing behavior without interception by the opponent player (Figs. 3,6). In the environment there are a learning agent (passer), a ball, an opponent, and two teammates (receivers). The problem here is to find the model which can most accurately describe the opponent's behavior from the viewpoint of the learning agent and to execute the policy which is calculated under the estimated model. It may take a time to distinguish the situation, therefore, we put an assumption : the opponent continues the one of its policies during one trial and changes after the trial.

3 A Multi-Module Learning System

Fig. 1 shows a basic architecture of the proposed system, that is, a multi-module reinforcement learning system. Each module has a forward model (predictor) which represents the state transition model, and a behavior learner (policy planner) which estimates the state-action value function based on the forward model in a reinforcement learning manner. This idea of combination of a forward model and a reinforcement learning system is similar to the H-DYNA architecture [Singh, 1992b] or MOSAIC [Doya et al., 2000]. The system selects one module which has the best estimation of a state transition sequence by activating a gate signal corresponding to a module while deactivating the gate signals of other modules, and the selected module sends action commands based on its policy.

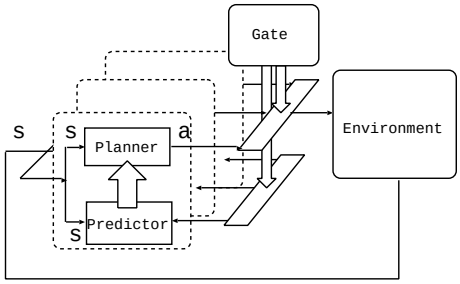


Figure 1: A multi-module learning system

3.1 Predictor

Each learning module has its own state transition model. This model estimates the state transition probability $\hat{\mathcal{P}}^a_{ss'}$ for the triplet of state s, action a, and next state s':

$$\hat{\mathcal{P}}^a_{ss'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \tag{1}$$

Each module has a reward model $\hat{\mathcal{R}}^a_{ss'}$:

$$\hat{\mathcal{R}}^a_{ss'} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \tag{2}$$

We simply store all experiences (sequences of state-action-next state and reward) to estimate these models.

3.2 Planner

Now we have the estimated state transition probabilities $\hat{\mathcal{P}}^a_{ss'}$ and the expected rewards $\hat{\mathcal{R}}^a_{ss'}$, then, an approximated state-action value function $Q(s, a)$ for a state action pair s and a is given by

$$Q(s,a) = \sum_{s'} \hat{\mathcal{P}}^a_{ss'} \left[\hat{\mathcal{R}}^a_{ss'} + \gamma \max_{a'} Q(s', a') \right] , \tag{3}$$

where $\hat{\mathcal{P}}^a_{ss'}$ and $\hat{\mathcal{R}}^a_{ss'}$ are the state-transition probabilities and expected rewards, respectively, and γ is a discount rate.

3.3 Module Selection

The reliability of the module becomes larger if the module does better state transition prediction during a certain period, else it becomes smaller. We assume that the module which does the best state transition prediction has the best policy against the current situation because the planner of the module is based on the model which describes the situation best. In our proposed architecture, the reliability is used for gating the action outputs from modules. We calculate the reliability g_i of the module i as follows:

$$g_i = \prod_{t=-T+1}^{0} e^{\lambda p_i^t}$$

where p_i is an occurrence probability of the state transition from the previous $(t-1)$ state to the current (t) one according to the model i, and λ is a scaling factor.

4 Task and assumption

The task of the learning agent is to pass the ball to one of the teammates while it avoids interception by the opponent. The game is like a three on one; there are one opponent and other three players. The player nearest to the ball becomes to a passer and passes the ball to one of the teammates while the opponent tries to intercept it.

Fig. 2 shows a mobile robot we have designed and built. Fig. 3 shows the simulator of our robots and the environment. The robot has an omni-directional camera system. A simple color image processing is applied to detect the ball area and an opponent one in the image in real-time (every 33ms). The left of Fig. 3 shows a situation in which the agent can encounter and the bottom right shows the simulated image of the camera with the omni-directional mirror mounted on the robot. The robot consists of an omni-directional vehicle of which motion (any translation and rotation on the plane) can be controlled.

Figure 2: A real robot

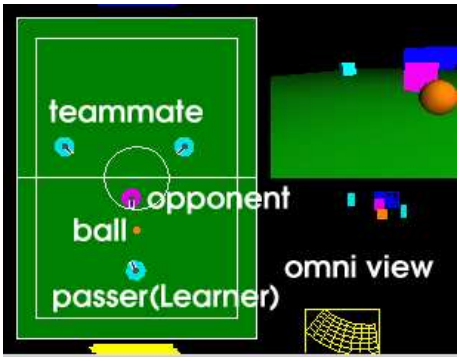
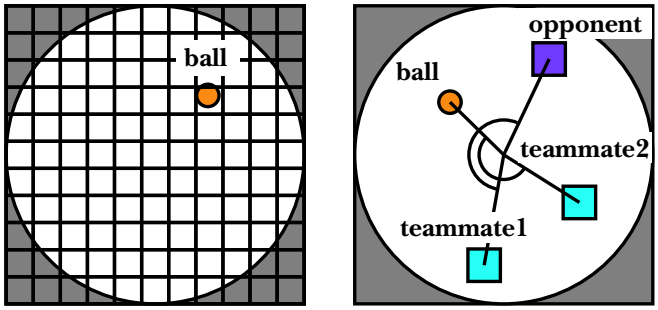


Figure 3: A simulation environment

The state space is constructed in terms of the centroid of the ball on the image, the angle between the ball and the opponent, and the angles between the ball and the teammates (see Figs. 4 (a) and (b)). We quantized the ball position space 11 by 11 as shown in Fig. 4 (a) and the each angle into 8. As a result, the number of state becomes $11^2 \times 8 \times 8 \times 8 = 61952$. The action space is constructed in terms of desired three velocity values (x_d,



(a) state variables (position) (b) state variables (angle)

Figure 4: State variables

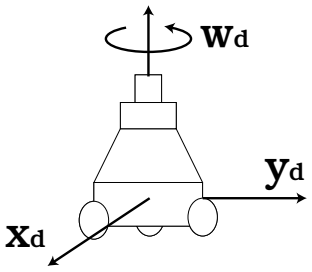


Figure 5: Action variables

y_d, w_d) to be sent to the motor controller (Fig. 5). Each value is quantized into three, then the number of action is $3^3 = 27$. The robot has a pinball like kick device, and it automatically kicks the ball whenever the ball comes to the region to be kicked. It tries to estimate the mapping from sensory information to appropriate motor commands by the proposed method.

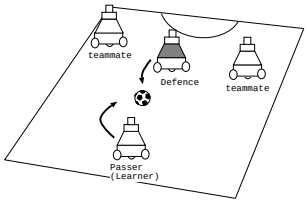
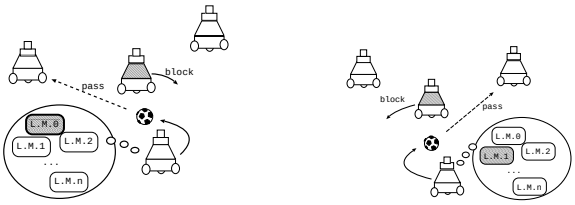


Figure 6: Task : 3 on 1

The initial positions of the ball, the passer, the opponent, and teammates are shown in Figs. 6. The opponent has two kinds of behaviors; it defend the left side, or right side. The passer agent has to estimate which direction the opponent will defend and go to the position in order to kick the ball to the direction the opponent does not defend. From a viewpoint of the multi-module learning system, the passer agent will estimate which situation of the module is going on, select the most appropriate module to behave. The passer agent acquires a positive reward when it approach to the ball and kicks it to one of the teammate dodging the opponent.



(a) right block (b) left block

Figure 7: Module switching

4.1 Learning Scheduling

We prepare a learning schedule composed of three stages to show its validity. The opponent fixes its defending policy as right side block at the first stage. After 250 trials, the opponent changes the policy to block the left side at the second stage and continues this for another 250 trials. Then, the opponent changes the defending policy randomly after one trial.

4.2 Simulation Result

We have applied the method to a learning agent and compared it with only one learning module. We have also compared the performances between the methods with and without the learning scheduling. Fig. 8 shows the success rates of those during the learning. The success indicates that the learning agent successfully kick the ball without interception by the opponent. The success rate indicates the number of successes in 50 trials. The multi-module system with scheduling shows a better performance than the one-module system. The "mono. module" in the figure indicates "monolithic module" system and it tries to acquire a behavior for both policies of the opponent. The monolithic module with scheduling means that we applied learning scheduling mentioned in **4.1** even though the system has only one learning module. The performance of this system is similar with multi-module system until the end of first stage (250 trials), however, it goes down at the second stage because the obtained policy is biased against the experiences at the fist stage and cannot follow the policy change of the opponent. Since the opponent takes one of the policies at random at the third stage, the learning agent obtains about 50% of success rate. "without scheduling" means that we do not applied learning scheduling and the opponent changes its policy at random from the start. Somehow the performance of the monolithic module system without learning scheduling is getting worse after the 200 trials. The multi-module system without learning schedule shows the worst performance in our experiments. This result indicates that it is very difficult to recognize the situation at the early stage of the learning because the modules has too few experiences to evaluate their fitness, then the system tends to select the module without any consistency. As a result, the system cannot acquires any valid policies at all. Fig. 9 shows an example sequence of the behavior when the agent executes its learned policy and the opponent behaves randomly after a fixed period. Figs. 10 and 11 show example sequences of the reliabilities while the opponent is blocking

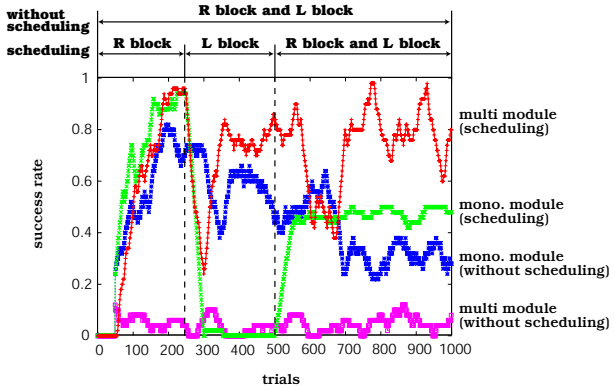


Figure 8: Success rate during the learning

the left and right sides. The "LM1" and "LM2" indecate the learning modules that are assigned the left and right block behaviors, respectively. The agent seems to fail to estimate the situation where the opponent is blocking left or right side at the beginning periods, however, the reliability of the appropriate module is getting higher after a few seconds and the agent successfully accomplished the task.

5 Conclusion and Future Work

In this paper, we proposed a method by which multiple modules are assigned to different situations which are caused by the alternation of the other agent policy and learn purposive behaviors for the specified situations as consequences of the other agent's behaviors. We have shown results of a simple soccer situation and the importance of the learning scheduling.

References

[Asada et al., 1996] Asada, M., Noda, S., Tawaratumida, S., and Hosoda, K. (1996). Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303.

[Asada et al., 1999] Asada, M., Uchibe, E., and Hosoda, K. (1999). Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence*, 110:275–292.

[Connell and Mahadevan, 1993] Connell, J. H. and Mahadevan, S. (1993). *ROBOT LEARNING*. Kluwer Academic Publishers.

[Doya et al., 2000] Doya, K., Samejima, K., ichi Katagiri, K., and Kawato, M. (2000). Multiple model-based reinforcement learning. Technical report, Kawato Dynamic Brain Project Technical Report, KDB-TR-08, Japan Science and Technology Corporation.

[Jacobs et al., 1991] Jacobs, R., Jordan, M., S, N., and Hinton, G. (1991). Adaptive mixture of local expoerts. *Neural Computation*, 3:79–87.

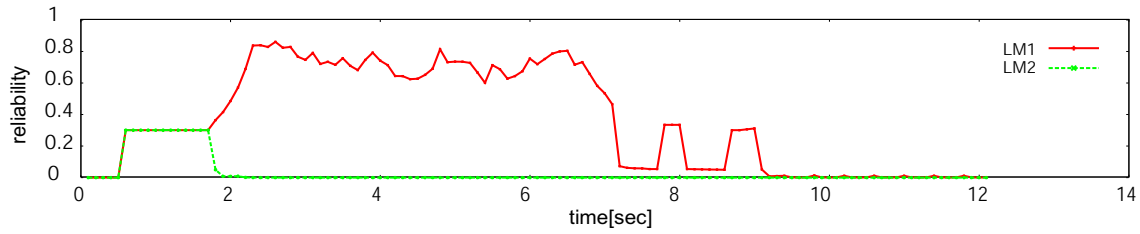


Figure 10: The sequence of the reliability signals of modules while the opponent is blocking the left side

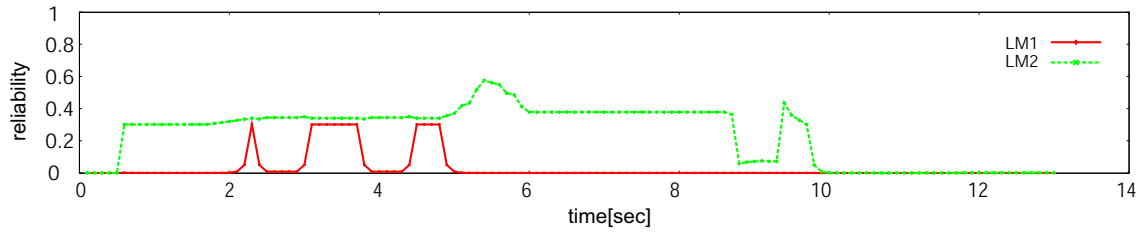


Figure 11: The sequence of the reliability signals of modules while the opponent is blocking the right side

[Singh, 1992a] Singh, S. P. (1992a). The effeicient learnig of multiple task sequences. In *Neural Information Processing Systems 4*, pages 251–258.

[Singh, 1992b] Singh, S. P. (1992b). Reinforcement learning with a hierarchy of abstract models. In *National Conference on Artificial Intelligence*, pages 202–207.

[Singh, 1992c] Singh, S. P. (1992c). Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8:323–339.

[Takahashi et al., 2002] Takahashi, Y., Edazawa, K., and Asada, M. (2002). Multi-module learning system for behavior acquisition in multi-agent environment. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages CD–ROM 927–931.

[Tani and Nolfi, 1997a] Tani, J. and Nolfi, S. (1997a). Self-organization of modules and their hierarchy in robot learning problems: A dynamical systems approach. Technical report, Technical Report: SCSL-TR-97-008.

[Tani and Nolfi, 1997b] Tani, J. and Nolfi, S. (1997b). Self-organization of modules and their hierarchy in robot learning problems: A dynamical systems approach. Technical report, Sony CSL Technical Report, SCSL-TR-97-008.

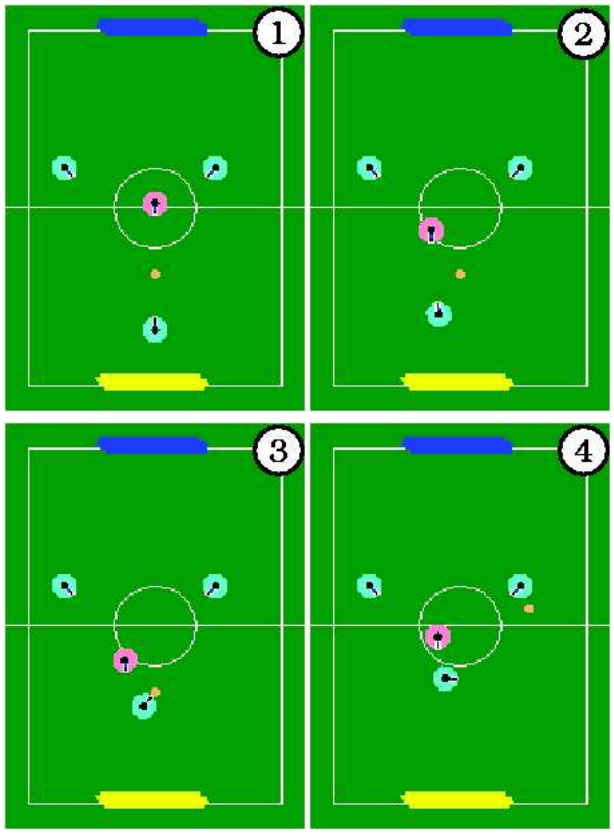


Figure 9: A sequence of a pass behavior while the opponent blocks the left side

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-0419-2 (5/1)

3台のカメラを用いた擬似ステレオによる3次元位置推定

Pseudo Stereo Method for Measuring 3D Position Using Three Cameras

清水彰一† **藤吉弘亘**† **長坂保典**† **高橋友一**††

Shoichi Shimizu, Hironobu Fujiyoshi, Yasunori Nagasaka, and Tomoichi Takahashi

† 中部大学, †† 名城大学

Chubu University, Meijo University

shiyou@vision.cs.chubu.ac.jp

Abstract

Multiple cameras have been used to get a view of a large area. In some cases, the cameras are layouted so that their views are overlapped to get a better view as a whole than non_overlapping layout. 3D information of the overlapping areas that are covered with two or three cameras can be get by stereo vision methods. Shifting shutter timings of cameras and using our pseudo stereo vision method, we can output 3D information faster than 30 fps.

In this paper, we propose a pseudo stereo vision method using cameras with different shutter timings. Using three cameras, two types of shutter timings are discussed. In three different shutter timings, 90 times of 3D position for a sec are obtained because proposed method can output 3D positions at every shutter timings of three cameras. In two different shutter timings, it is possible to calculate the 3D position at 60 fps with better accuracy.

1 はじめに

ロボットが行動するためには，センサによる環境認識・理解技術から得られる情報を基に行動プランニングを行い，ロボットを制御するフィードバック機構が必要である．視覚センサでロボットを迅速にかつ正確に制御するためには，ビジュアルフィードバックのサイクルを高速に行う必要があり，特に，カメラから得られる画像から有用な情報を得るための高速なビジョンシステムが不可欠となる．その際には，ロボット等の対象とする物体の3次元位置をより正確に検出する必要がある．ロボカップにおいても，ループシュートを打つサッカーロボット [Muratec, 2003] が登場しており，その際には，ボールの軌跡をサッカーフィールド上の2次元位置ではなく，3次元位置を高速に求める必要がある．

ビジョンシステムの高速化として，ロボカップ小型リーグでは，60 fps の高速カメラが用いられている [R.D'Andrea, 1999, 加藤, 2002]．これは，60 fps のカメラを用いたダブルバッファリングによる高速処理を行うものである．しかし，このシステムでは，1台のカメラを使用しているため，対象物の3次元位置を求めることができない．

対象物の3次元位置を推定する手法としてマルチベースラインステレオが既に提案されており，多様な分野で用いられている[Okutomi, 1993]．一般にステレオ視の際，運動している対象物の3次元位置を正確に求めるためには各カメラの画像を同時に取得する必要がある．一方，複数の非同期カメラを用いた対象物の距離画像を求める手法として，Zhou らによる距離画像の生成法[Zhou, 2003] が提案されている．2台の非同期カメラを用いた距離画像の生成法では，存在しない画像を非同期カメラによって発生するシャッタータイミングの時間ずれとオプティカルフローを用いて自動生成し，生成した画像と実画像の対応より，距離画像を取得する．時間ずれは，基準画像上に各カメラ2点ずつ，過去4点の特徴点を求め，その比率により得られる．基準画像に存在しない特徴点の基準画像への投影には，エピポーラ直線と基準画像上の2特徴点により構成される直線の交点を求めることにより行う．Zhou らの手法の出力は，時刻 $t-1$(1 フレーム前) の出力であり，最新フレームでの出力ではないという問題がある．

我々は，2台の非同期カメラを用いた3次元位置推定法を提案している [Shimizu, 2004]．2台のカメラ間のシャッタータイミングのずれを利用し，60 fps で対象物の3次元位置を得ることが可能である．本報告では，3台のカメラを用いた際に考えられる2種類のシャッタータイミングにおける3次元位置推定法について提案する．

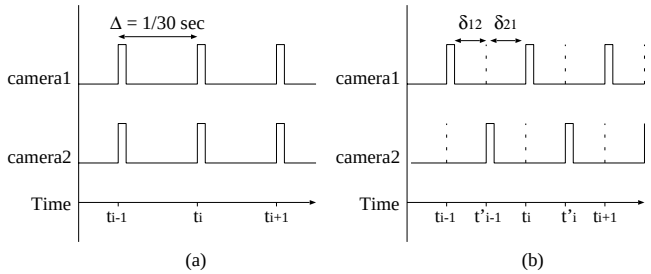


Figure 1: 同期・非同期のシャッタータイミング

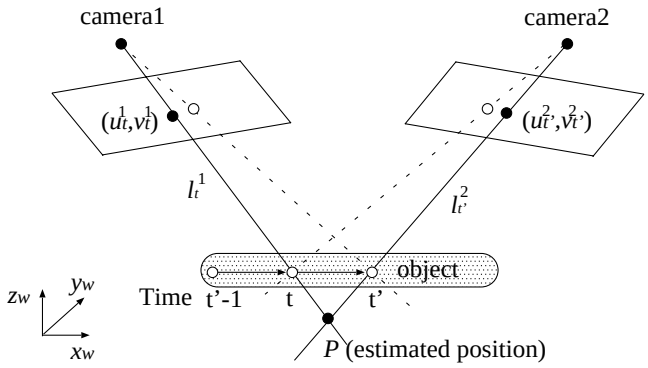


Figure 2: ステレオ視における時間ずれによる誤差

2 章では，2 種類のシャッタータイミングとその手法について，3 章では，シミュレーションによる移動物体の運動復元について実験を行い，提案手法の有効性を示す．4 章では，実際の実験において，高速に円運動する対象物の 3 次元位置推定の実験を行い，提案手法の有効性を示す．

2 複数のカメラを用いた 3 次元位置推定

ステレオ視では対象物の 3 次元位置を求めるために，2 枚の画像を同時刻で取得する必要がある．一般的なステレオ視では，通常のカメラ (30 fps) を用いた場合，3 次元位置の出力は最大 30 fps となる．2 台のカメラによるステレオ視におけるカメラのシャッタータイミングを図 1(a) に示す．カメラ間が同期していない場合，各カメラ間に時間ずれが生じる (図 1(b))．最新画像がカメラ 2 の場合，(u_t^1, v_t^1) と 時刻 $t' = t + \delta$ の対応点 $(u_{t'}^2, v_{t'}^2)$(時間ずれを含んだ対応点) を用いてステレオ視により高速に移動する対象物の 3 次元位置を求めると，図 2 に示すように誤差が生じる．

これに対し，我々は，非同期カメラを用いた 3 次元位置推定法を提案している [Shimizu, 2004]．シャッタータイミング毎に 3 次元位置を求めることにより，早いサイクルでの 3 次元位置の出力を得ることができる．以下に，3 台のカメラを用いた 2 種類のシャッタータイミングにおける 3 次元位置推定法について示す．

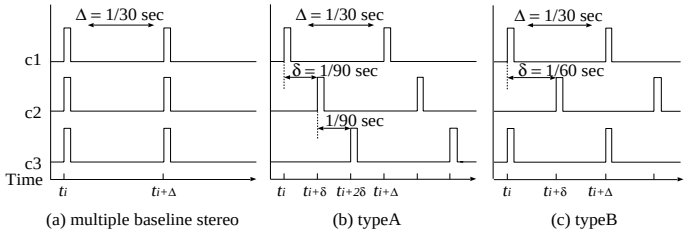


Figure 3: シャッタータイミングの組み合わせ

2.1 カメラ 3 台におけるシャッタータイミング

3 台のカメラを用いた場合，考えられる 3 種類のシャッタータイミングの組み合わせを図 3 に示す．図 3(a) は，マルチベースラインステレオシステムにおけるシャッタータイミングである．本報告では図 3(b) を typeA，(c) を typeB とし，この 2 種類のシャッタータイミングにおける 3 次元位置推定について検討する．typeA の場合，各カメラ間の時間ずれ Δ を 1/90 秒と設定する．この場合，シャッタータイミング毎に 3 次元位置を求めるため 90 fps の出力を得ることが可能となる．一方，typeB では，カメラ 1 とカメラ 3 を同期するように設定し，ステレオ視により 3 次元位置を求める．カメラ 2 が最新フレームの際は，前 2 フレームにおいてステレオ視により求めた結果から，線形予測を用いて 3 次元位置を求め，光線情報により 3 次元位置を修正し出力を得る．また，カメラ 2 のシャッタータイミングは，カメラ 1，3 より 1/60 秒ずれているため，3 次元位置を 60fps で得ることが可能となる．以下に 2 種類のシャッタータイミングにおける 3 次元位置推定法について述べる．

2.2 TYPE-A: 3 つのシャッタータイミング (90 fps)

typeA では，3 台の各カメラ間のシャッタータイミングを 1/90 秒ずつずらし，この時間ずれを考慮して各シャッタータイミング毎に 3 次元位置推定を行う．以下に 3 台のカメラを用いた 3 次元位置推定の手法を示す．

Step1 前 2 フレームの 3 次元位置を計算

Step2 線形予測により最新フレームの 3 次元位置を計算

Step3 最新フレームの画像面上の点を通る光線情報による 3 次元位置の修正

各カメラのシャッタータイミング毎に，上記の Step1〜3 により 3 次元位置を求める．従って，本手法は 1 秒間に 90 ポイントの 3 次元位置を出力することが可能となる．以下に各 Step における処理について示す．

2.2.1 前 2 フレームの 3 次元位置計算

図 4 に前 2 フレームの 3 次元位置計算について示す．各カメラ画像より対象物の 2 次元画像座標 (u, v) を求める．カメラ 1 によって撮影された物体の画像座標を (u_t^1, v_t^1),

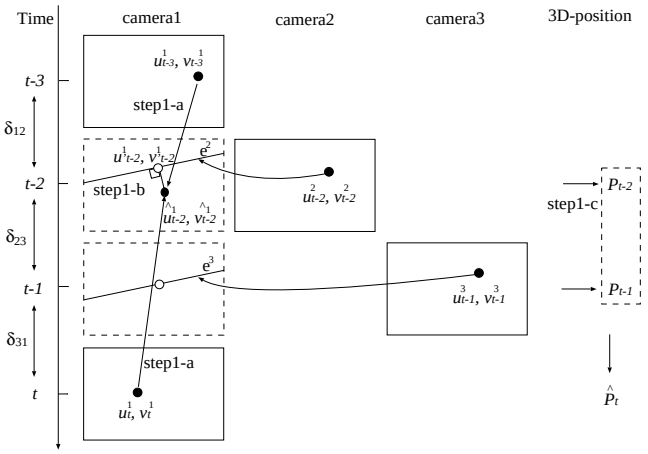


Figure 4: 前 2 フレームの 3 次元位置計算

カメラ 2 の画像座標を (u_t^2, v_t^2)，カメラ 3 の画像座標を (u_t^3, v_t^3) とする．画像座標は，キャプチャ終了時の時刻を基に時系列の順に並べられる．本来ならば，ステレオ視の際，同時刻 t の対応点を求める必要があるが，提案手法における 3 台のカメラ間は $\delta(\delta = \delta_{12} = \delta_{23} = \delta_{31} = 1/90$ 秒) の時間ずれを持つ．そのため，図 4 に示すように時刻 $t-1$ におけるカメラ 3 の画像座標 (u_{t-1}^3, v_{t-1}^3) に対応するカメラ 1 の画像座標は存在しない．そこで，以下に示すアルゴリズムを用いて対応する時刻におけるカメラ 1 上の画像座標 $(\hat{u_{t-1}^1}, \hat{v_{t-1}^1})$ を求め，3 次元位置 $\boldsymbol{P}_{t-1}$ をステレオ視により計算する．

Step1-a カメラ 1 において，前後フレームにおける実際に観測された 2 点の画像座標より $t-1$ における画像座標 $(\hat{u_{t-1}^1}, \hat{v_{t-1}^1})$ を次式により推定する．

$$\begin{aligned} \hat{u_{t-1}^1} &= \frac{(\delta_{12}+\delta_{23})u_t^1 + \delta_{31}u_{t-3}^1}{\delta_{12}+\delta_{23}+\delta_{31}}, \\ \hat{v_{t-1}^1} &= \frac{(\delta_{12}+\delta_{23})v_t^1 + \delta_{31}v_{t-3}^1}{\delta_{12}+\delta_{23}+\delta_{31}} \end{aligned} \quad (1)$$

Step1-b 時刻 $t-1$ におけるカメラ 3 の観測された画像座標点 (u_{t-1}^3, v_{t-1}^3) から対応するカメラ 1 上のエピポーラ線 e^3 を求める．Step1-a で計算した $(\hat{u_{t-1}^1}, \hat{v_{t-1}^1})$ が最も近いエピポーラ線上の点 $(u_{t-1}'^1, u_{t-1}'^1)$ を $t-1$ におけるカメラ 1 の画像座標とする．

Step1-c カメラ 1 上の推定した画像座標 $(u_{t-1}'^1, v_{t-1}'^1)$ から世界座標空間の直線 $\boldsymbol{l}_{t-1}^1$ とカメラ 3 上の観測された (u_{t-1}^3, v_{t-1}^3) より直線 $\boldsymbol{l}_{t-1}^3$ を求め，その交点を $t-1$ における対象の 3 次元位置 $\boldsymbol{P}_{t-1}$ とする．

以上の処理を $t-2$ においても行い，前 2 フレームの 3 次元位置を求める．

2.2.2 線形予測による最新フレームの 3 次元位置推定

既に計算された前 2 フレームの結果 $\boldsymbol{P}_{t-1}$ と $\boldsymbol{P}_{t-2}$ を用いて，図 5 に示すように，最新フレーム t における予測位置 $\hat{\boldsymbol{P}}_t = [x_w, y_w, z_w]^T$ を次式より求める．

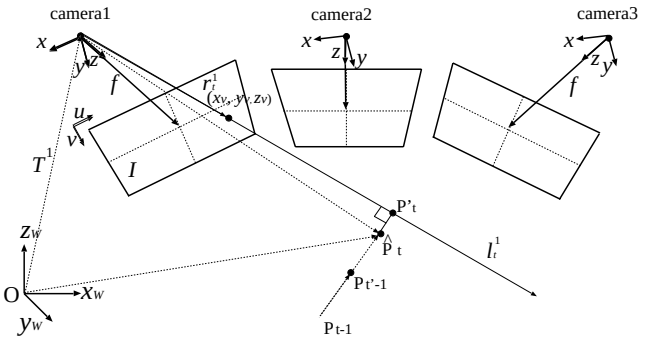


Figure 5: 光線情報による 3 次元位置推定

$$\hat{\boldsymbol{P}}_t = \boldsymbol{P}_{t-1} + \boldsymbol{v}_{t-1}\delta, \quad \boldsymbol{v}_{t-1} = \frac{\boldsymbol{P}_{t-1} - \boldsymbol{P}_{t-2}}{\delta} \quad (2)$$

式 (2) は，前 2 フレームより速度を線形に求めたものである．このとき，最新フレームはカメラ 1 の結果とする．また，過去の計算結果を用いた位置予測には拡張カルマンフィルタを用いたもの[Browning, 2002, 金子, 1996]や 3 次元曲線が考えられるが，本報告では観察する時間間隔が 1/90 秒と短いことから線形に求める．

2.2.3 光線情報による 3 次元位置の修正

最新フレーム t におけるカメラの画像座標から求まる 3 次元空間の直線を求める．最新フレーム t における結果がカメラ 1 の場合，世界座標における原点 O からカメラ 1 への平行移動ベクトルを $\boldsymbol{T}^1 = [Tx, Ty, Tz]^T$，物体の画像座標 (u_t^1, v_t^1) から求められた世界座標における直線 $\boldsymbol{l}_t^1$ の傾きを表すベクトルを $\boldsymbol{r}_t^1 = [x_v, y_v, z_v]^T$ とすると，図 5 に示す光線は次式に示す直線 $\boldsymbol{l}_t^1$ と表現できる．k は実数である．

$$\boldsymbol{l}_t^1 = k\boldsymbol{r}_t^1 + \boldsymbol{T}^1 \quad (3)$$

本来ならば，式 (2) により求めた予測位置 $\hat{\boldsymbol{P}}_t$ はこの直線 $\boldsymbol{l}_t^1$ 上に存在する．しかし，予測位置 $\hat{\boldsymbol{P}}_t$ は過去に計算した位置から推定したものであり予測誤差を含んでいる．そのため，予測位置 $\hat{\boldsymbol{P}}_t$ は図 5 に示すように必ずしも最新フレームの結果より得られる直線 $\boldsymbol{l}_t^1$ 上に存在するとは限らない．

そこで，予測位置 $\hat{\boldsymbol{P}}_t$ が垂直に交わる直線上の点 $\boldsymbol{P}_t'$ を求める．$\boldsymbol{P}_t'$ は予測位置 $\hat{\boldsymbol{P}}_t$ の直線 $\boldsymbol{l}_t^1$ の傾きを表すベクトル $\boldsymbol{r}_t^1$ 方向への正射影ベクトルであり，次式で求めることができる．

$$\boldsymbol{P'}_t = \frac{(\hat{\boldsymbol{P}}_t - \boldsymbol{T}^1)\cdot \boldsymbol{r}_t^1}{|\boldsymbol{r}_t^1|^2}\boldsymbol{r}_t^1 + \boldsymbol{T}^1 \quad (4)$$

これにより，線形予測で求めた $\hat{\boldsymbol{P}}_t$ に最も近いカメラ 1 上の画像座標点 (u_t^1, v_t^1) を通る直線上の点 $\boldsymbol{P'}_t$ を対象の最新フレーム t による 3 次元位置とする．カメラ 2，カメ

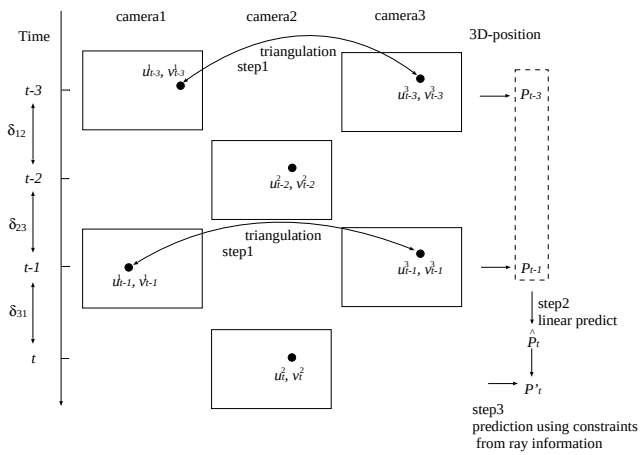


Figure 6: 3 次元位置推定 (typeB)

ラ 3 が最新フレームの場合も画像座標 (u_t^2, v_t^2)，(u_t^3, v_t^3) から各直線 $\boldsymbol{l}_t^2$，$\boldsymbol{l}_t^3$ を求め，上記の処理を同様に行う．

2.3 TYPE-B: 2 つのシャッタータイミング (60fps)

線形予測を用いて，より精度の良い 3 次元位置を求めるためには，前 2 フレームにおける 3 次元位置を正確に求める必要がある．図 3(c) の場合，2 台のカメラ (カメラ 1 とカメラ 3) のシャッタータイミングは同期しているため，ステレオ視により 3 次元位置を求めることができる．図 6 に示すようにステレオ視で得られた前 2 フレームの 3 次元位置 $\boldsymbol{P}_{t-1}$ と $\boldsymbol{P}_{t-3}$ を用いて，最新フレームの 3 次元位置 $\boldsymbol{P}'_t$ を線形予測により推定する．その後，typeA と同様に，光線情報の制約を用いて 3 次元位置を修正する．

3 次元位置は，2 台の同期したカメラによるステレオ視の結果と，1 台のカメラから得られる光線情報による修正を加えた結果が交互に出力される．この場合，1 秒間に typeA(90 ポイント) より少ない 60 ポイントの出力となる．

3 シミュレーション実験

3 次元空間の対象物の運動を復元するシミュレーション実験により，提案手法の評価を行う．

3.1 対象物の運動復元

対象物が仮想世界座標空間 (3,000 × 2,000 × 2,000 mm) を移動していると仮定し，その運動復元を行う．3 台のカメラは，高さ 3,000 mm に平行となるように設置してあるとする (図 7 参照)．このとき，3 次元空間内の対象物の運動として，以下に示す 3 種類の等速・非等速運動を対象とする．

- 等速運動 (直線): $(x, y, z) = (3{,}000,\ 1{,}200,\ 0)$ から $(x, y, z) = (0,\ 1{,}200,\ 2{,}000)$ に向けて直線上を速度 3,000 mm/sec で移動
- 等速運動 (螺旋): $(x, y) = (1{,}000,\ 1{,}000)$ を中心に半径 620 mm，角速度 4.7 rad/s で螺旋上を移動

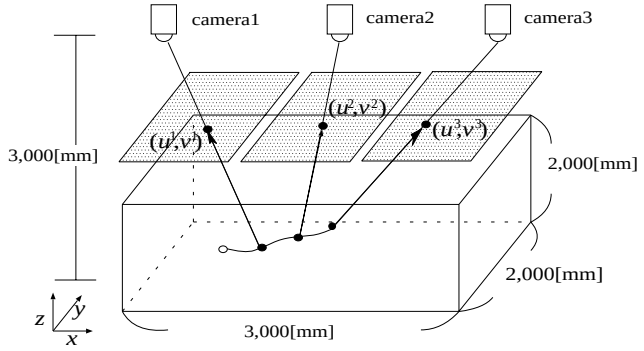


Figure 7: 仮想カメラ配置

- 非等速運動: 高さ 2,000 mm の位置からボールを落下させたときの放物運動 (重力加速度 g=9.8 m/s^2)

これらの対象物の運動軌跡を，各カメラの仮想画像平面へ投影する．シミュレーション実験では，各カメラの仮想画像平面上の点 (u, v) を用いて，提案手法により 3 次元位置を推定する．また，本実験おいて画像平面上での観測誤差はない．

3.2 実験結果

表 1 に各運動を復元した結果と真値との推定誤差を示す．表 1 において，typeA のシャッタータイミングにおける非同期は，図 2 に示す時間ずれを含んだ対応点を用いたステレオ視の結果である．typeA では，本手法 (typeA) の結果が非同期より良いことは明らかである．

typeB のシャッタータイミングにおける線形予測は，同期した 2 台のカメラからステレオ視により前 2 フレームの 3 次元位置を求め，その 2 点から線形予測した結果である．提案手法と線形予測を比較すると，提案手法の精度が良いことが分かる．これは，線形予測した結果から，光線情報を用いた修正を行うことにより位置推定精度が向上しているためである．

Table 1: 3 次元位置の推定値と真値との平均誤差 [mm]

シャッタータイミング		fps	等速		非等速
			直線	螺旋	
typeA	非同期	90	24.6	20.8	14.8
	提案手法	90	1.1	2.0	1.7
typeB	線形予測	60	0.2	1.4	4.4
	提案手法	60	0.2	0.5	1.5

4 複カメラによる評価実験

本提案手法を，1 台の PC と 3 台のカメラを用いて実装し，等速円運動する対象物の 3 次元位置を求める実験により評価を行う．

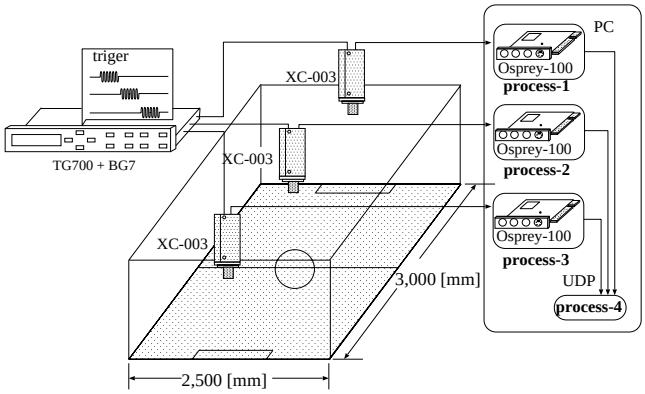


Figure 8: ビジョンシステムの概略

<機器構成>

- PC (DELL PRECISION 530)
  - CPU (XEON DUAL PROCESSOR 2.2 GHz
  - MEMORY (1.0 GB)
- カメラ (SONY XC-003) 3台
- FRAME GRABBER (ViewCast Osprey-100) 3台
- TV信号発生装置 (TG700 + BG7型)

Figure 9: 機器構成

4.1 本システムの概略

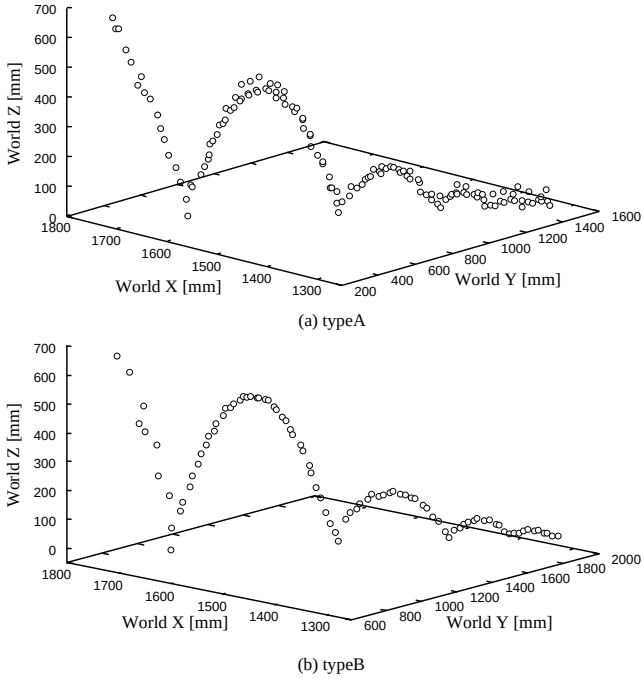
3 台のカメラを用いたビジョンシステムの概略を図 8 に示す．カメラは高さ 2,800 mm の位置に床面上の 2,000 × 3,000 mm の領域が視野に入るように設置した．それぞれのカメラは世界座標 (x_w, y_w, z_w) と画像座標 (u, v) により，キャリブレーション済みである [Tsai, 1987]．カメラからの映像信号は，1 台の PC にインストールされた 3 枚のフレームグラバに入力され，それぞれの画像が取り込まれる．また，TV 信号発生装置を用いて，各カメラに NTSC 信号を各タイプ毎にシャッタータイミングを発生させ入力する．本システムの機器構成を図 9 に示す．

それぞれのカメラからの画像は，独立したプロセス process-1，process-2，process-3 で各シャッタータイミング毎に処理される．各プロセスは，画像処理により得られたカメラ i における対象物の画像座標 (u^i, v^i) と，その処理した画像のキャプチャ終了時刻を UDP 通信により 3 次元位置を計算する process-4 に送信する．

4.2 実験

実際にボールを投げたとき 1.5 秒間の復元結果を図 10 に示す．図 10(a) では，135 ポイントのボールの 3 次元位置を出力しており，90 fps のカメラと同等の出力であることが分かる．図 10(b) では，90 ポイントのボールの 3 次元位置を出力しており，60 fps のカメラと同等の出力であることが分かる．

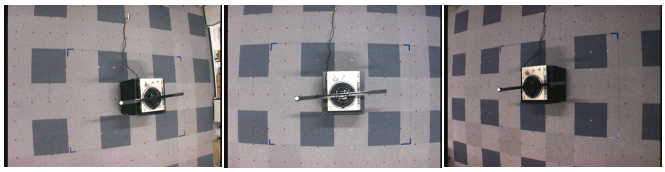
本手法の評価を行うため，図 11 に示すようにターンテーブルとゴルフボールを用いて実験を行う．長さ 1,000



(a) typeA

(b) typeB

Figure 10: 3 次元位置推定結果



(a) camera1 (b) camera2 (c) camera3

Figure 11: 実験の様子

mm の物差しの端にボールを固定し，ターンテーブルを回転させることで半径 500 mm の等速円運動となる．ターンテーブルは高さ 500 mm の箱に置かれており，ボールの床からの高さは 660 mm となる．ターンテーブルは 45 rpm のスピードで回転しており，角速度は $(45 \times 2\pi)/60 = 0.478$ rad となる．

4.3 実験結果

typeA と typeB の 2 種類のシャッタータイミングによる 3 次元位置推定結果の z_w 軸方向における平均と分散を表 2 に示す．共に，z_w 軸の平均 (ボールの高さ) は，実際の高さ 660[mm] から 5 [mm] 以内であった．typeB の分散値は typeA より低く，シミュレーション結果と同様に typeB の精度が高いことが分かる．

Table 2: z 軸の平均と分散

	平均 [mm]	分散
typeA	664.5	2143.7
typeB	662.3	112.0

4.4 まとめ

本報告では，2 種類のシャッタータイミングに応じた 3 次元位置推定法を提案し，その評価を行った．提案手法は，3 台のカメラ全てのシャッタータイミングが 1/90 秒ずれている場合 (typeA)，各シャッタータイミング毎に 3 次元位置を求めることにより，1 秒間に 90 ポイントの出力を得ることが可能である．一方，同期した 2 台のカメラと 1/60 秒ずれたシャッタータイミングのカメラの場合 (typeB)，60 fps で 3 次元位置を得ることができ，より正確に求めることが可能である．

ロボカップ小型リーグでは，2004 年から，フィールドサイズが大きくなり，1 台のカメラではフィールド全体を映すことは不可能であると考えられる．本手法は，カメラ間で重なる位置が在れば，より早く 3 次元位置を求めることが可能である．

謝辞

本研究は，栢森情報科学振興財団の助成を受けて遂行された．

参考文献

[Muratec, 2003] Muratec FC: http://ww.muratec.net/robot/

[R.D'Andrea, 1999] R. D'Andrea, et al: Detailed vision documentation, http://robocup.mae.cornell.edu/

[加藤, 2002] 加藤恭佑, 日比野晋也, 児玉幸司, 村上和人, 成瀬正: ロボカップのための高速な小型ロボット検出法について, 情報処理学会研究報告, Vol.CVIM-136, No.16, pp.115-122, 2003.

[Okutomi, 1993] M. Okutomi, and T. Kanade; A Multiple Baseline Stereo, In Proc. IEEE Trans, PAMI, Vol. 15, No. 4, pp. 353-363, 1993.

[Shimizu, 2004] S. Shimizu, H. Fujiyoshi, Y. Nagasaka and T. Takahashi: A Pseudo Stereo Vision Method for Unsynchronized Cameras, ACCV2004, Vol.1, pp.575-580, 2004.

[Zhou, 2003] C. Zhou and H. Tao: Dynamic Depth Recovery from Unsynchronized Video Streams, In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.351-358, 2003.

[Bruce, 2000] J. Bruce, T. Balch and M. Veloso: Fast and Inexpensive Color Image Segmentation for Interactive Robots, IROS-2000, Vol.3, pp.2061-2066, 2000.

[Browning, 2002] B. Browning, M. Bowling and M. Veloso: Improbability Filtering for Rejecting False Positives, In Proc. IEEE International Conference on Robotics and Automation, pp.120-200, 2002.

[金子, 1996] 金子俊一，堀内一仁，本多庸悟: 多重カルマンフィルタによる 3 次元運動推定, 信学論, Vol.J79-D-II, No.5, pp.840-850, 1996.

[Tsai, 1987] R. Y. Tsai: A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses, In IEEE Journal of Robotics and Automation, Vol.RA-3, Num.4, pp.323-344, 1987.

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-0419-3 (5/1)

進化型計算を用いたチーム戦略獲得手法の一提案

A Proposal of a Method for Acquiring Team Strategy using Evolutionary Computation

中島 智晴，高谷 将裕，有働 昌代，石渕 久生
Tomoharu NAKASHIMA, Masahiro TAKATANI, Masayo UDO, and Hisao ISHIBUCHI
大阪府立大学大学院 工学研究科
College of Engineering, Osaka Prefecture University
{nakashi, takatani, hisaoi}@ie.osakafu-u.ac.jp

概 要

This paper proposes an evolutionary method for acquiring team strategy of RoboCup soccer agents. The action of an agent in a subspace is specified by a set of action rules. The antecedent part of action rules includes the position of the agent and whether the nearest opponent agent is near the agent or not. The consequent part indicates the action the agent has to take when the antecedent part of the action rule is satisfied. The action of each agent is encoded by a bit string that represents the action rules. A chromosome is the concatenated string of bit strings for all the agents. The main genetic operator in our evolutionary method is mutation where a value of each bit is changed with a prespecified probability. Through computer simulaitons, we discuss the results and show some future researches.

1 はじめに

RoboCup はサッカーロボットによる世界選手権で，その最終目標は「2050 年までに FIFA ワールドカップのチャンピオン・チームに勝つことができる完全自律型のヒューマノイド・ロボット・チームを作ること」である．協調マルチエージェント問題を主に取り扱う RoboCup では，ソフトコンピューティングに基づいた手法がよく適用されている [1].

サッカーエージェントの開発には，パスやドリブルなどのような低レベルの行動に対する実装と，意思決定や強調プレイなどのような高レベルの行動に対する実装が必要である．中島ら [1] は，低レベルの行動としてボールインターセプト問題を取り上げ，ファジィQ学習によりオンラインでスキルの上達を行う手法を提案している．

一方，進化型計算を用いた戦略の進化の研究も盛んに行われている．例えば，Chellapilla ら [2, 3] は，進化的プログラミングの枠組みを用いて，チェッカーゲームを行う強いエージェントを専門家の知識を使用することなしに自動生成するアルゴリズムについての詳細な議論を行っている．

また，RoboCup サッカーエージェントへの進化型計算の応用は，Luke ら [4] をはじめ，数多くある．Luke らは，共進化の概念を取り入れた遺伝的プログラミングを用いて進化初期段階では団子サッカーであったチーム戦略が，進化の終了段階では，フォーメーションサッカーを行うような戦略を獲得できる事が示されている．

本論文では，高レベルの行動実装を進化型計算により行う手法を提案する．具体的には，進化型計算により，各エージェントがボールを持っているときの行動が決定される．この行動は，エージェントの存在する位置や状態も考慮に入れる．数値実験の結果から，世代が進むにつれてサッカーエージェントが 効果的な行動や戦略を獲得することを示す．

2 問題設定

本研究では，進化型計算を用いることにより，良いサッカーのチーム戦略を獲得することが出来るかどうかを調べる．そのために，対戦する敵チームを一定に設定し，その対戦チームに対して有効な戦略を獲得する事が出来るかどうかを調べる事にする．

戦略の進化的獲得を行うサッカーチームの振る舞いを以下に示す．各サッカーエージェントは行動ルールを保持しており，ボールを持っているときにはその行動ルールに従って行動を行う．ボールを持っているかどうかの判断は，ボールとエージェントの距離が kickable area 以内かどうかで行うものとする．ここで，kickable area はサッカーサーバ内で定められている正の定数である．

行動ルールは以下の形式の If-Then ルールである．

$$R_j:\ \text{If Agent is in Area } A_j \quad \text{and the nearest opponent is } B_j \quad \text{then the action is } C_j \tag{1}$$

ここで，R_j はルールのインデックス，A_j はエージェントの位置に関する条件部の値で 1 ～ 48 の整数値をとる．この整数値は図 1 のように 48 分割されたサッカーフィールドの中の部分領域を示している．B_j はエージェントから最も近い敵エージェントがエージェントに非常に近いかどうかを示す条件部で near もしくは not near が使用される．near か not near かの判断はエージェントと敵エージェントとの距離が kickable are×1.5 以内であれば near，そうでなければ not near とする．C_j は結論部行動を表す整数であり，1 ～ 10 の整数値をとる．各々の整数値は以下の行動を表している．

1. 敵ゴール方向に向かってフィールドの x 軸方向と並行にドリブルする．
2. 敵ゴールの中心（両ポストの間）に向かってドリブルをする．
3. 敵ゴールの中心に向かってドリブルをするが，敵を避けることに細心の注意を払う．
4. 最近傍の味方エージェントに向かってパスをする．最近傍の味方エージェントが自分より後ろに存在する場合には敵ゴール方向にクリアボールを蹴る．
5. ２番目に近い味方エージェントに向かってパスをする．2 番目に近い味方エージェントが自分よりも後ろに存在する場合は敵ゴール方向に向かってクリアボールを蹴る．
6. サッカーフィールドの内側に向かってボールをクリアする．
7. サッカーフィールドの外側に向かってボールをクリアする．
8. 敵ゴール前にボールをキックする．
9. 最近傍の味方エージェントにスルーパスを行う．
10. 敵ゴールに向かってシュートする．

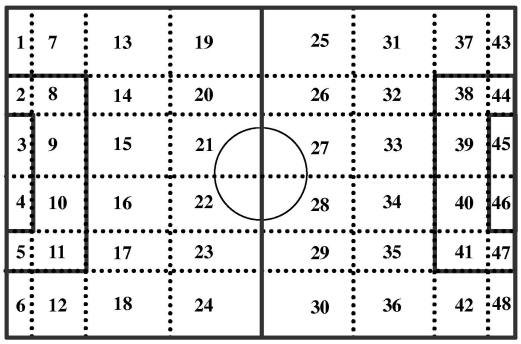


図 1: フィールドの分割

ただし，エージェントが敵ゴール側ペナルティエリア内にいる場合（図 1 のインデックス 38 ～ 41 と 44 ～ 47）には，まず，シュートコースが空いているかどうかを確認し，シュートが可能かどうかを調べる．シュートが可能であると判断すればシュートをし，シュートが可能でなければ行動ルールの結論部に従って行動を行う．

ボールを持っていないエージェントの振る舞いは以下の通りである．まず，自チームのエージェントの中で，どのエージェントが最も早くボールに到達することができるかをボールの位置と速度，およびエージェントの位置と速度から計算する．最も早くボールに到達することができるエージェントは最大の速度でボールに向かう．最も早くボールに到達することができると判断されなかったエージェントはそのエージェントのホームポジションとボールを結ぶ線分上に自分がいるように動く．

3 進化手法によるチーム戦略の獲得

本節では，チーム戦略の獲得のための進化的手法について説明する．チーム戦略は文字列で表現され，その文字列を進化的操作により修正して強いチーム戦略を探索する．

3.1 文字列の生成

2 節で説明したように，エージェントがボールを持っているときの行動は式 (1) の形式で記述される行動ルールに従う．サッカーフィールドは 48 個の部分領域に分割されており（図 1 参照），かつ，最近傍の敵エージェントが near か not near かの判断が式 (1) の条件部に含まれていることを考慮すると，各々のエージェントは $48 \times 2 = 96$ 個の行動ルールを持つことになる．本研究では，キーパーを除いた 10 エージェントの行動ルールを進化的手法により獲得するので，一つのチームがもつ戦略は $96 \times 10 = 960$ 個の行動ルールからなる集合となる．本研究では，これら 960 個の行動ルールは固定とし，結論部行動が求めるパラメータとなる．従って，進化的手法の目的は，より良いチーム戦略を表現する 960 個の行動ルールの結論部を探索することである．チームをあらわす文字列は，長さ 960 の文字列である．図 2 に，各々のエージェントに対する行動ルールの文字列を示す．

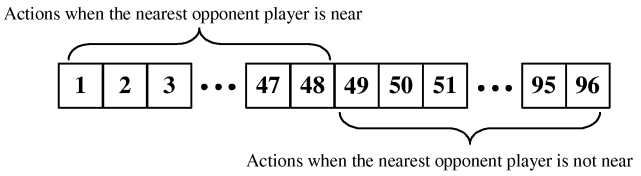


図 2: 1 つのエージェントに対して用いられる文字列

図 2 で，最初の 48 文字は最近傍の敵エージェントが対象となるエージェントに近い時の行動ルールを，次の 48 文字は，最近傍の敵エージェントが対象となるエージェントに近くない時の行動ルールを表している．チーム戦略は，図 2 の文字列を 10 エージェント分を連結させた 960 文字からなる．各ビットには 1 ～ 10 の整数値が入り，各整数の値は，2 章で示されているエージェントの行動を表す.

3.2 文字列の評価

進化型計算の基本的なアイデアは，個体の評価値が高いものをなるべく保存し，次の世代の生成のために活用する事である．本研究では，個体，すなわちチーム戦略の評価指標として，実際にサッカーを行わせたときの獲得得点を考えることにする．獲得した得点が高ければ高いほど，その個体の評価値は高い．もし，複数の個体の獲得得点が同じである場合には，第二評価指標として，失点を考えることにする．この場合，失点が少ないチームほどそのチームの評価が高くなる．ただし，獲得得点にすでに差がある場合には，失点は全く考慮しないものとする．

3.3 進化操作

本研究で提案する進化型計算では，交叉操作は用いられない．突然変異操作のみを用いてチーム戦略を表す文字列に修正を加えることにする．

突然変異操作では，各ビットに対して，あらかじめ決められた突然変異確率で値を変更する．突然変異を受けるビットはランダムに 1 ～ 10 から選択された値に置き換えられる．置き換える前と同じ値が選ばれることも許されている．

世代更新は以下の方法により行われる．まず，現在の個体群に含まれる個体からランダムに個体を選択し，その個体に突然変異操作を施す事により新しい個体を生成する．あらかじめ決められた数だけ新しい個体を生成した後，現在の個体群と新しく生成された個体集合の中から評価値が高いものを個体群サイズだけ抜き出し，それを次の世代の個体群とする．

以上，本研究で提案する進化型計算の手続きをまとめると以下のようになる．

Step 1: 初期個体群の生成．ランダムもしくはある決められた方法により初期個体群に含まれる個体を個体群サイズだけ生成する．

Step 2: 現在の個体群からランダムに個体を選択し，突然変異操作を施す事により新しい個体をあらかじめ決められた数だけ生成する．

Step 3: 現在の個体群に含まれる個体と Step 2 で新しく生成された個体の評価を行う．本研究では，個体の評価はサッカーの対戦による成績を元にしているため，ある個体に対して常に同じ評価が行われるとは限らないことに注意する．

Step 4: 評価の良い個体から個体群サイズ分の個体を次世代の個体群とする．

Step 5: 終了判定．終了条件が満たされていれば手続きを終了する．満たされていない場合は Step 2 へ移動する．

4 数値実験

4.1 実験の設定

本章では，本研究で提案した進化型計算により，良いチーム戦略が獲得できることを数値実験により確認する．数値実験で用いられた進化型計算のパラメータは以下のとおりである．

- 個体群サイズ 1
- 新しく生成する個体数 1
- 突然変異確率 1/96
- 初期個体の生成 Hand-coding

個体群サイズを 1 としたのは，個体の評価に時間がかかるため（1 試合は 10 分で構成される），出来るだけ個体群サイズを小さくする必要があったためである．数値実験を加速させるために，初期個体は Hand-coding により生成した．

個体を評価するための対戦として，以下の 2 通りを考えた．1 つは，現在の個体群に含まれる個体と新しく生成された個体を直接対決させる方法である．この場合，試合に勝利したチームの文字列が次世代の個体群を構成する個体として取り扱われる．もう 1 つは，ある固定したチームとの対戦結果により個体の評価を行う方法である．本研究では，UvA Trilearn2003 のベースコードを使用した．すなわち，この方法ではまず，現在の個体群に含まれる個体と UvA Trilearn2003 とを対戦させる．次に，新し

く生成された個体と UvA Trilearn2003 とを対戦させる．二つの対戦成績を比較し，獲得得点が多いチームを次世代の個体群に属する個体とする．獲得得点が同一である場合には，失点の小さいチームの評価をより良いとする．失点も同じであった場合には，現在の個体群に含まれる個体がそのまま次世代の個体群を構成するものとする。

なお，本研究で進化の対象としたチームは UvA Trilearn2003 のベースコードを元にして２章で述べたような行動方式を取るチームに修正する事により作成された．

4.2 実験結果

まず，直接対決により個体の評価・世代更新を行った場合の数値実験結果を示す．進化型計算を 6000 世代まで実行し，それにより得られた 1000 世代目，2000 世代目，3000 世代目，4000 世代目，5000 世代目，6000 世代目の個体と初期世代を 10 回対戦させてその成績を調べた．表 1 に数値実験結果を示す．

表 1: 数値実験結果その１

世代	勝数	負数	引分	平均得点	平均失点
1000	4	2	4	0.9	0.7
2000	4	5	1	4.5	4.8
3000	7	3	0	4.6	3.9
4000	5	5	0	4.3	3.5
5000	2	6	2	3.3	4.5
6000	5	4	1	3.8	4.3

表１より，第 1000 世代に得られた個体は平均得点が 0.9 と低いのに対して，それ以降の世代で得られた個体は１試合平均３点以上となっている．このことにより，進化型計算により，より高い攻撃力をもつチームの戦略が獲得されたといえる．ただし，本研究で提案している進化型計算手法では，守るためのチーム戦略は考慮されておらず，攻撃するためのチーム戦略に焦点が置かれているために，平均失点は，世代が進むにつれて大きくなっているという結果も得られた．守るためのチーム戦略の進化を考えることは，今後の研究課題の一つである．

次に，個体の評価をある一定のチーム（UvA Trilearn2003 ベースチーム）に対する対戦成績とした場合の進化型計算を 600 世代実行した．100 世代目，300 世代目，500 世代目，600 世代目で得られた個体を，UvA Trilearn2003 ベースチームと 10 回対戦させたときの数値実験結果を表 2 に示す．

表 2 より，世代が進むにつれて，平均得点が増加し，また，個体の勝利数が増加していることが分かる．また，第 100 世代で得られた個体の平均失点が 0.7 であったのに対して，第 600 世代で得られた個体の平均失点は 0.5 と減少

表 2: 数値実験結果その２

世代	勝数	負数	引分	平均得点	平均失点
100	2	4	4	0.3	0.7
300	1	4	5	0.1	0.4
500	2	3	5	0.2	0.4
600	4	2	4	0.8	0.5

もしている事が分かる．このことから，進化型計算により，個体の評価のために使用されたチームに対して有効であるチーム戦略が獲得できたといえる．

5 おわりに

本研究では，進化型計算を用いて RoboCup サッカーシミュレーションリーグにおけるサッカーチーム戦略の進化を行う手法の提案を行った．サッカーエージェントは，行動ルールの集合によって決められた行動に従って行動し，進化型計算は行動ルールの最適な結論部行動を探索する．進化型計算では，交叉操作は行わず，突然変異操作のみを用いて個体を表す文字列を修正する．個体の評価は，サッカーの試合による対戦成績で行われた．数値実験の結果，提案した進化型計算手法によりサッカーチームがより良い戦略を獲得出来る事が示された．

今後の研究方向としては，個体数を増やして個体群の多様性を増加させることの獲得されるチーム戦略への影響を調べる事，守るためのチーム戦略を獲得する手法の提案，異なる複数のチームとの対戦から個体を評価し，どのようなチームに対しても柔軟に対応できるチーム戦略を獲得する事が出来るような手法の提案などが挙げられる．

参考文献

[1] 中島智晴，有働昌代，石渕久生,“ファジィ Q 学習によるサッカーエージェントの行動獲得”，日本知能情報ファジィ学会誌, Vol.15, No.6, pp.702–707, (2003).

[2] K. Chellapilla, and D.B. Fogel, “Evolving Neural Networks to Play Checkers Without Relying on Expert Knowledge”, *IEEE Trans. on Neural Networks*, Vol.10, No.6, pp.1382–3191, 1999.

[3] K. Chellapilla, and D.B. Fogel, “Evolving an Expert Checkers Playing Program Without Using Human Expertise”, *IEEE Trans. on Evolutionary Computation*, Vol.5, No.4, pp. 422–428, 2001.

[4] S. Luke and L. Spector, “Evolving Teamwork and-Coordination with Genetic Programming”, *Proceedings of the First Annual Conference on Genetic Programming*, pp.150–56, 1996.

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-0419-4 (5/1)

センサ履歴に基づく多自由度ロボットの環境に適した行動選択

A Behavior Selection Method based on Sensor history for Humanoid Robots

宮下敬宏 今川拓郎 石黒浩

Takahiro MIYASHITA, Takuro IMAGAWA and Hiroshi ISHIGURO

ＡＴＲ知能ロボティクス研究所 大阪大学

ATR Intelligent Robotics Labs. and Osaka University

miyasita@atr.jp

Abstract

In this paper, we propose a behavior selection method based on sensor history. The robot's motion, the environment and the history of the sensors have close and important relationship. Thus, we make a map which denotes the relationship between them by analyzing the sensor history and apply it to the robot. The robot can decide own motion adapted to the environment by utilizing the decision tree based on the map. We verify the validity of our method by performing preliminary experiments with a small humanoid type robot, named Robovie-M.

1 はじめに

ヒューマノイドロボットのように多自由度を持つロボットの行動を自動的に生成することは，ロボットの持つ自由度が多く動作の探索範囲が広いこと，環境がロボットに与える物理的な制約が多いことから，困難である．また，ヒューマノイドロボットがどのように動作するかは，ユーザからの要求によって決定されることが多く，動作する前から決まっていることが多い．従って，ヒューマノイドロボットに要求される第一の目標は，あらかじめ決められた動作を環境に適応させることである．

従来，ロボットの動作選択のための環境認識手法としては様々なものが提案されてきた．具体的には，車輪型移動ロボットの動作のための超音波距離センサと視覚センサを併用して認識するもの[前山, 1994]や，３次元モデルと視覚による認識に基づくもの[Fennema, 1987]などがある．また，ヒューマノイドロボットの動作を環境に適応させるものに関しては，反射的なものが多く[Hirose, 2001]，複雑な観測あるいは長期的な観測によってわかる弾性や摩擦などの環境の特性などに関しては，動作前にあらかじめ限定しているものが多い．

ロボットがセンサによって環境を認識する能力には限界があり，また床面上の弾性や摩擦などは，実際に動作してみないと認識できない．また，多くの種類のセンサを使って複雑なセンサ処理を行い環境を認識する手法は，実際に環境を動くロボットに搭載することが困難であり現実的ではない．さらに，動作前にセンサによって認識する過程において，ロボットが誤認識をした場合，その動作を実行することが困難となるだけでなく，ロボットのセンサによって適切な動作に修正することは原理的に困難となる．

そこで本研究では，ロボットが実際に行動するときのセンサ情報と環境の特徴をマッピングし，ロボットの行動とセンサ履歴から環境を判別する判別木をマップに基づいて作成し，ロボットが実際に環境内を失敗を繰り返しながら動きまわることで，その環境に適した動作選択を行う手法を提案する．具体的には，まず，摩擦力や弾性の違う複数の床面上でヒューマノイドロボットの歩行や起きあがり動作などの基本動作をユーザが作成する．次に，その動作時のセンサ出力の時系列データを蓄える．蓄えた時系列データは，動作名と環境特徴をインデックスとしてデータベース化する．データ利用時には，まず環境判別のための探索木をデータベースに基づいて作成しておき，ロボットにユーザから基本動作指令が与えられたとき，まずその動作を行う．そのとき獲得されるセンサ出力の時系列データから，その環境に適した動作かどうかを，探索木に従って調べる．適していない場合は，動作と観測を繰り返し，最終的に環境に適した動作を選択する．

提案手法は，動作と環境が反映されるセンサ履歴を利用するため，多くのセンサを必要としない．本稿では，実際に小型ヒューマノイドロボットに搭載されている２軸加速度センサのみで環境認識をする基礎実験を行い，提案手法の実現可能性とその有効性を示す．

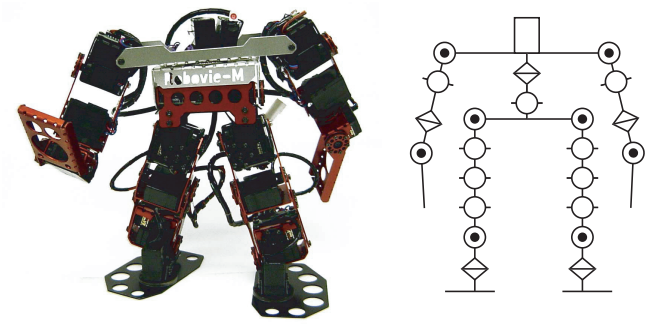


Figure 1: Overview and DOFs arrangement of Robovie-M

2 Robovie-M

本稿での提案手法は，ロボットが動作時に得られるセンサ出力の時系列データによって環境の判別をおこなう．そのため，ロボットは環境に対して多くの自由度を持つ方が，ある環境での様々な動作に対するセンサ出力を得ることができるため，環境の識別能力を高くすることができる．また，本手法では，動作を行い失敗することを繰り返すことで適した行動を選択する手法であるため，行動が失敗した際にロボットが損耗しないことと，失敗した状態から復帰できることも重要である．

本研究では，様々な動作を実現できる自由度と転倒に関する頑健性を持ち，転倒後の復帰動作が原理的に可能な小型ヒューマノイドロボットとして株式会社国際電気通信基礎技術研究所知能ロボティクス研究所で開発したRobovie-M(製造販売：ヴイストン株式会社) を用いる．本ロボットの全体図を Fig. 1 に示す．Robovie-M は，Fig.1 に示すとおり，各脚に 6 自由度，各腕に 4 自由度，胸部に 2 自由度の合計 22 自由度を持つ小型ヒューマノイドロボット（身長 290mm，重量 1.9kg）であり，各関節はすべて模型用サーボモジュールで構成されている．また，センサとしては，胸部に 2 軸加速度センサを搭載しており，ロボットの前後方向と左右方向の 2 軸の加速度を検出することができる．動作制御ボードとしては H8 CPU ボードを胸部に搭載しており，この CPU ボードのシリアルポートを介して外部コンピュータと接続することができる．

このロボットの動作は，外部コンピュータ上の動作作成ソフト “Robovie Maker” によってロボットを動作させながらリアルタイムに作成することができ，完成した動作はロボットの CPU ボードにダウンロードして用いることで，外部コンピュータと接続せずに自立的に動作させることができる．ロボットの動作例は`http://www.vstone.co.jp/top/p_info/robot/robovie-m.html` の動画に示す．動画に示されるように，転倒した際には加速度センサを用いて転倒方向を検出し，復帰動作を自動的に行うことができる．

3 センサ履歴に基づく動作選択

3.1 ［センサー動作－環境］の対応付けマップの構築

以下に提案手法の概要を示す．まず，［センサー動作－環境］の対応付けマップを以下の手順で作成する．

1. ロボットが活動する実際の環境で，その環境に適した動作を作成する．
2. 作成した動作を行っているときのセンサ出力時系列データを蓄積する．
3. 蓄積したセンサ出力時系列データを FFT によって周波数領域に写像し，その周波数スペクトルパターンと，それに対応する動作と環境の名称を関連づけて［センサー動作－環境］の対応付けマップの要素候補とする．
4. 既にあるマップの要素と，手順 3 の要素候補とを，動作名称と環境名称で比較し，同じものがあればセンサ出力を平均化し，一つの要素としてまとめる．同じものがなければ新しい要素としてマップに登録する．

上記の手順を，ロボットが活動するすべての環境で繰り返し，マップの要素を作っていく．ここで，作成する動作はロボットに求められる動作であり，与えられるタスクによって異なるが，移動に関する動作と，転倒状態などの動作を失敗した状態から復帰するための動作は常に用いる．Fig. 2 に，この手順によってできあがるマップの概略図を示す．

3.2 環境判別木による行動選択

3.1 節で述べたマップから，環境を判別し，その環境に適した動作を選択するための環境判別木を作成する．作成する方法は，マップの要素となっているセンサ出力の周波数スペクトルパターンを多次元ベクトルとして扱い，そのベクトルにセンサ出力に関連づけられている動作ラベルを別次元として付け加えたベクトルに基づいて，環境を決定するための決定木作成を行う．作成手法には C4.5[Quinlan, 1993]を用いる．ID3[Duda, 2001]などの別の決定木作成手法でも可能である．

作成した決定木は，以下のように使用する．

1. ロボットは，ユーザーあるいは外部プログラムからの指令により，基本動作のなかのいずれかの動作を行う．この動作は，現在の環境に適しているかどうかはわからない．
2. 動作時にセンサ情報を取得し，その時系列データを蓄積する．
3. 時系列データが 256 点以上蓄積できれば，そのデータをノイズ除去の後に FFT によって周波数領域に写像する．

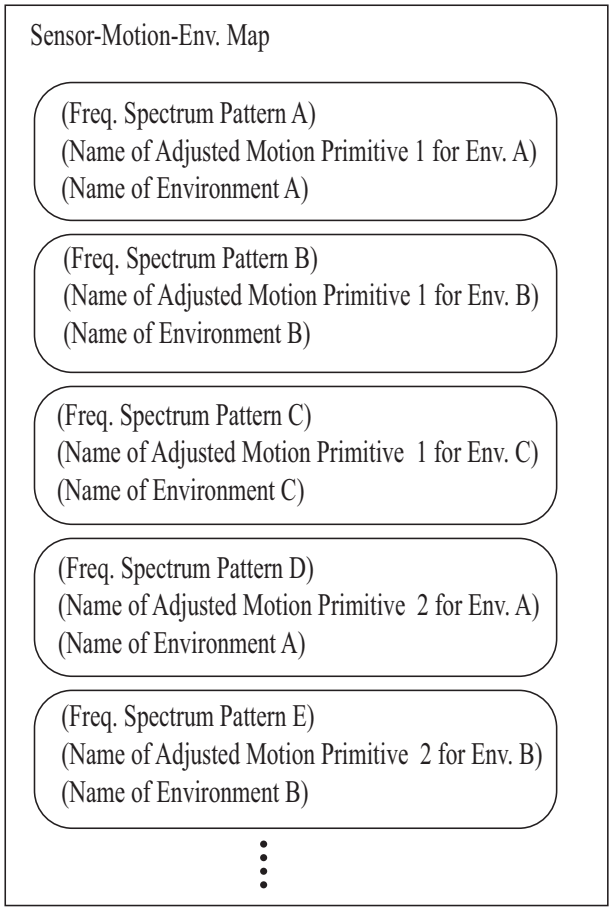


Figure 2: Outline of the Sensor-Motion-Environment Map

4. 周波数スペクトルパターンを表した多次元ベクトルと動作ラベルに基づいて決定木を探索する．

5. 探索後のノードが環境を表した葉ノードである場合は探索を終了する．ノードが葉ノードではない場合は，決定木に従い次の動作を選択する．(2) へ．

また，例外処理として，蓄積中に転倒などで明示的に動作が失敗した場合は，選択していた動作は不適切な動作であると判断し，強制的に失敗から回復するための基本動作を選択し，上記の手順 (2) に進む．

4 実験

［センサー動作－環境］対応付けマップを構築するための基礎実験として，小型ヒューマノイドロボット Robovie-M による動作生成実験を行い，動作時のセンサ出力の解析を行った．本章では，実験結果を示すとともに本手法が原理的に可能であることを示す．

4.1 基本動作生成

Robovie-M の基本動作として，ここでは以下の 3 つの動作を生成する．

1. 歩行動作

2. 仰向けの状態から脚を振り下ろす反動で起きあがる動作

3. 俯せの状態から起きあがる動作

これらの動作を，条件の異なる 3 つの整地平面上で作成する．整地平面の条件は，以下の通りである．

1. スポンジ（厚さ 10mm）で構成される柔軟な床面．

2. ロボカップのフィールドで使用されているカーペット（厚さ 2mm）で構成される床面．柔軟性は小さいが摩擦が大きい．

3. プラスチック板で構成される床面．柔軟性はなく，摩擦が小さい．

動作の作成は，ユーザが動作作成ソフト “Robovie Maker” を用いて行う．

実際に作成した動作を Fig.3 に示す．Fig.3(a),(b) が，それぞれスポンジの上での歩行，反動起きあがり動作を示している．基本動作にかかる時間は環境の違いによって変化はせず，1 周期の歩行が 1.7[sec]，反動起きあがり動作が 3.3[sec]，俯せの状態から起きあがる動作 6.7[sec] となっている．スポンジの弾性は測定していないが，歩行動作時の脚先の沈み込み量は 4mm であった．

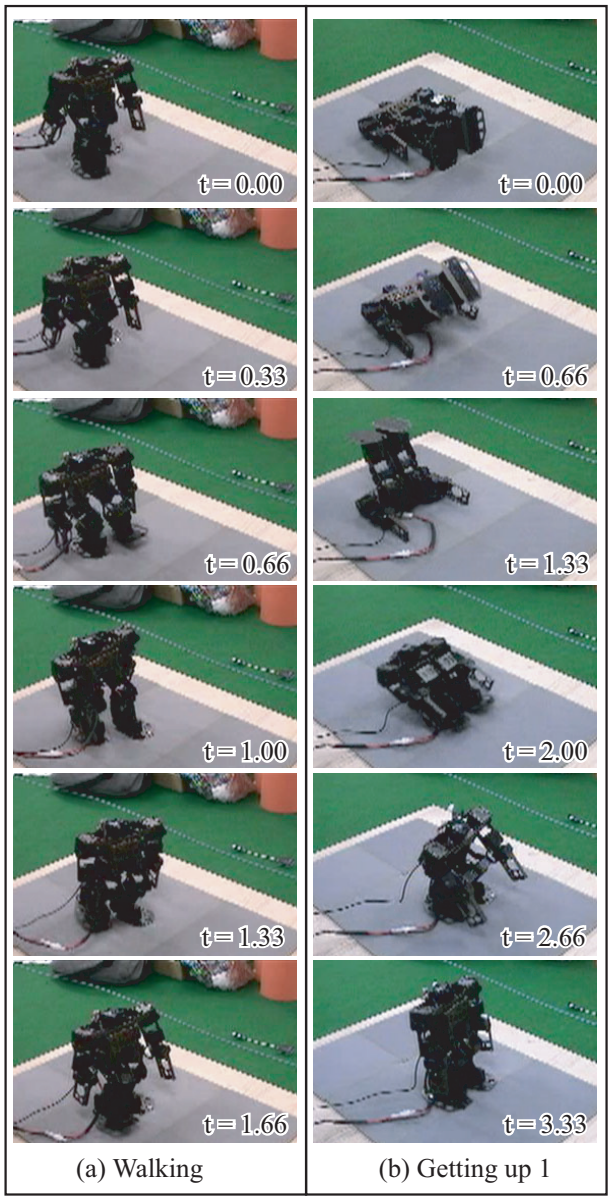


Figure 3: Examples of motion primitives on the soft plain

4.2 ［センサー行動－環境］マップのためのセンサ出力解析

上記の各環境に適した基本動作を繰り返し行い，動作中のセンサ出力を蓄積し，マップを構築する．ここでは，センサ出力として小型ヒューマノイドロボットの胸部に搭載されている 2 軸加速度センサを用いて，その出力と基本動作の組み合わせで環境が識別できることを示す．

本実験では，4.1 節で述べた 3 種類の床面（スポンジ，カーペット，プラスチック板）上で 3 つの基本動作（歩行・仰向けからの反動起きあがり・俯せからの起きあがり）を行い，そのときのセンサ出力として加速度センサ出力を 20[Hz] でサンプリングした．サンプリングしたセンサ出力は，3 点平均フィルタによって電気的・機械的なノイズを除去し，そのデータの個数にあわせた FFT によって周波数領域に変換する．各環境で，それぞれの環境に適した歩行動作を行った際の周波数スペクトルを Fig.4 に示す．歩行動作は 8 歩行い，280 点のデータであるため 256 点の FFT を行った．

Fig.5 にはスポンジ床面で，プラスチック板の床面に適した歩行動作を行った際のセンサ出力を示す．このグラフと，プラスチック板の床面で，プラスチック板の床面に適した歩行を行った際のセンサ出力のグラフ (Fig.4(a))，およびスポンジ床面で，スポンジ床面に適した歩行を行った際のセンサ出力のグラフ (Fig.4(c)) を比較すると，それぞれで周波数スペクトルの分布が異なっていることがわかる．

Table.1 は，センサ出力の再現性と判別可能性を検討するために，各センサ出力の散らばり度合いを示した．すなわち，3 つの環境で，3 つの環境に適した歩行動作を行った際に得られるセンサ出力の周波数スペクトルを 128 次元のセンサ出力ベクトルとし，各環境，各動作の組み合わせで作られるセンサ出力ベクトル 9 個の間のユークリッド距離の偏差を求めた．また，環境と動作の条件を同じくして動作実験を 5 回繰り返してセンサ出力ベクトルを獲得し，そのセンサ出力ベクトル間のユークリッド距離の偏差を求めた．表からわかるように，環境と行動が同じであれば各センサ出力の再現性は高く，また条件が異なる場合は，距離が離れるためクラスタリングは可能である．

Table 1: Standard deviations of sensor output vectors

	standard deviation
different conditions	0.49
same conditions (average)	0.15

最後に，Fig.6 には，歩行動作について異なる条件で得られる 9 個のセンサ出力ベクトル間の距離を用いて，重心法によって作られる樹木図を示す．この樹木図から，たとえば，スポンジ床面上で，プラスチック板上の床面に適した歩行動作を行った場合と，スポンジ床面上で，カーペット床面に適した歩行動作を行った場合とでは，区別がしにくいことがわかる．この場合，歩行動作とは別の基本動作を組み合わせることで，環境に適した行動を選択することが可能となる．

5 おわりに

本研究では，ロボットの行動とそのとき観測されるセンサの履歴，およびロボットが活動する環境のそれぞれが密接な関わりを持っていることを利用し，ロボットが実際に行動するときのセンサ情報と環境特徴のマップから，環境に適した行動を選択するための手法を提案した．また，基礎実験によって，本手法の要となるロボットの動作とセンサ情報と環境のマップが作成可能であることを示した．

あらかじめロボットが経験している環境であれば，本手法によって，適した動作を選択することが可能となる．基礎実験で用意した 3 つの床面は，弾性と摩擦がそれぞれ大きく異なる環境であり，整地平面においては弾性と摩擦を考慮することで広範な環境に対処することが可能となる．今後の課題は，小型ヒューマノイドロボットへの本手法の動作選択部分の実装である．

謝辞

本研究は独立行政法人情報通信研究機構 (NICT) の研究委託により実施したものである．また，ヴイストン株式会社および大阪大学石黒研究室の高山氏に謝意を表する．

参考文献

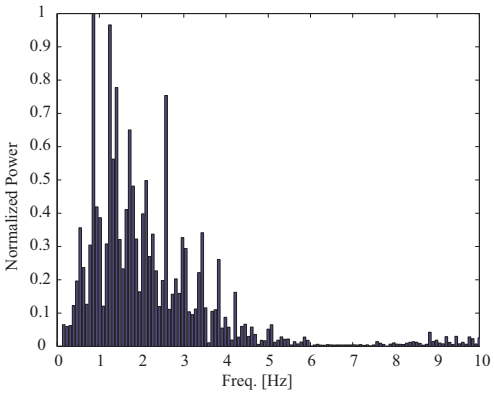
[前山, 1994] 前山, 大矢, 油田: 街路樹をランドマークとしたポジショニングによる移動ロボットの屋外走行実験, 日本機械学会ロボティクス・メカトロニクス講演会'94 講演論文集, pp. 305–308, 6, 1994.

[Fennema, 1987] Fennema C., Hanson A., Riseman E., Beveridge J. and Kumar R.: Model-Directed Mobile Robot Navigation, IEEE Trans. on Sytems, Man and Cybernetics, Vol. 20, No. 6, pp. 1978–1984, 1987.

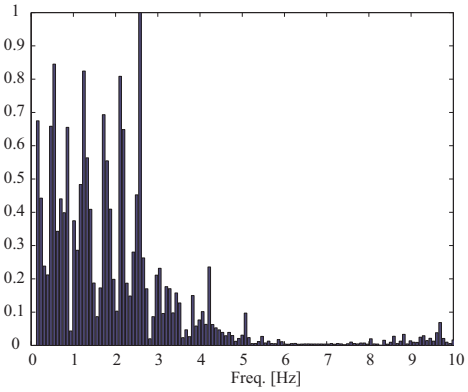
[Hirose, 2001] Hirose M., Haikawa Y., Takenaka T. and Hirai K.: Development of Humanoid Robot ASIMO, Proc. Int. Conf. on Intelligent Robots and Systems, Workshop2, 2001.

[Quinlan, 1993] Quinlan J. R.: *C4.5: Programs for Machine Learning,* Morgan Kaufman, 1993.

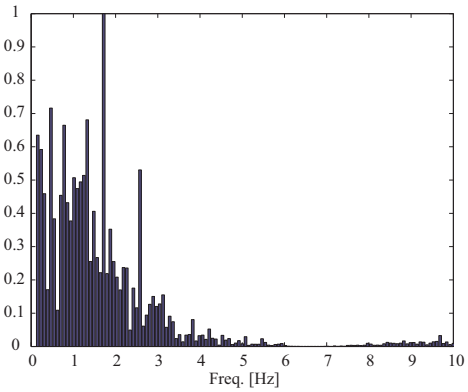
[Duda, 2001] Duda R. O., Hart P. E. and Stork D. G.: *Pattern Classification second edition,* John Wiley & Sons, 2001.



(a) Freq. spectrum pattern for Walking motion on plastic slick plain



(b) Freq. spectrum pattern for Walking motion on carpet plain



(c) Freq. spectrum pattern for Walking motion on sponge plain

Figure 4: Examples of freq. spectrum pattern of adjusted motion primitives

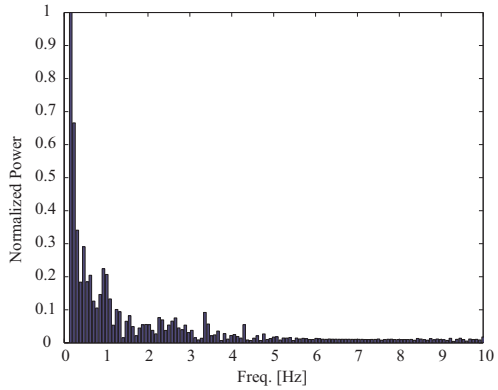


Figure 5: Freq. spectrum pattern for non-adjusted Walking motion on sponge plain

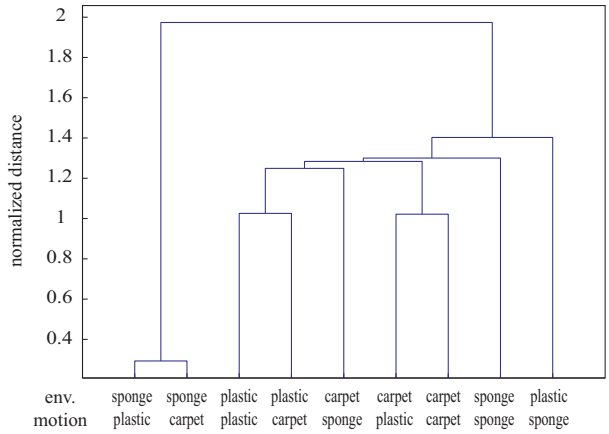


Figure 6: Dendrogram of walking motions under all conditions

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-0419-5 (5/1)

Skill Learning for a Humanoid to Pass a Ball

Masaaki Kikuchi[1], Masaki Ogino[1] and Minoru Asada[1,2]
[1]Dept. of Adaptive Machine Systems,
[2]HANDAI Frontier Research Center,
Graduate School of Engineering, Osaka University,
Osaka, Japan
Email: {kikuchi, ogino}@er.ams.eng.osaka-u.ac.jp,
asada@ams.eng.osaka-u.ac.jp

***Abstract*— This paper proposes a method of bahavior generation for humanoids in which a robot learns sensorimotor maps in each motion module as the forward and inverse relationships between optic flows in the robot's view and motion parameters. Humanoids use these maps to determine appropliate motion parameters that generate a desired flow given by a planner. Each module consists of a planner and sensorimotor maps of primitives and can accomplish a simple task. Using a predefined module transition rule, humanoids can accomplish a complex task. Passing a ball (face-to-face pass) between two humanoids which have different camera lens and body parameters is realized as an example task.**

I. Introduction

Understanding the causal relationship between a self-induced motion and changes in sensory data is essential for a robot to move around and manipulate objects in the environment. This relationship can be obtained by sensorimotor mapping from the self-induced motion and the seonsory data that reflect the consequecen of the motion. Vision is one of the typical sensations to percieve what is going on in the external world, and has been utilized to obtain the relationship of sensorimotor mapping. The methods can be classified into two categories. One is a top-down approach in which a designer constructs vision and motor systems separately and gives their relationship, that is, the sensorimotor mapping. In this approach, each system is calibrated in the reference frame of the global coordinate system [4] [5]. These methods do not consider changes in an environment or in robot morphology (body size and structure, and sensor configuration). Therefore, if such changes happen, the designer need to give another sensorimotor mapping or to devise an adaptive model both of which seem difficult to prepare or to build since it seems impossible for the designer to prepare all kinds of relationships in advance.

On the other hand, the other approach does not calibrate each system separately, but directly correlate visual information and motor commands. In this approach, an optic flow has been used for mapping, because an optic flow caused by the motion contains the information of the causal relationship between the environmental changes and the motion. The mapping between optic flows and motion commands are learned for obstacle avoidance planned by the learned forward model [2] or finding obstacles that show different flows from the environments using reinforcement learning [3] in wheel-based robots. Also, it is used for object recognition by active touching by the upper torso of the humanoid [1]. In these studies, the robots have much fewer DoFs than humanoids, therefore it seems difficult to directly apply their methods to realize various kinds of humanoid behaviors. One solution is to decompose a humanoid action into basic motion primitives and to acquire the sensorimotor mapping in each motion primitive. This makes it possible for a robot to learn the sensorimotor mapping on line because in each motion primitive the relationship between the motion parameters and the sensor values is usually much simpler than in case of multiple motions.

This paper presents a method of visuo-motor learning for behavior generation of humanoids, and, as an example task, passing a ball between two humanoids (face-to-face pass) is realized based on the sensorimotor mappings of motion primitives. The task is decomposed into three basic motion modules: trapping a ball, approaching to a ball and kicking a ball to the opponent. Each motion module can be further decomposed into several motion primitives, each of which has motion parameters to control the motion trajectory. The sensorimotor mapping is learned as the forward and inverse relationships between these motion parameters and optic flow information in each motion. The acquired sensorimotor maps are used to select the appropriate motion primitive and its parameters to realize the desired pathway or destination in the robot's view given by the planner.

The rest of the paper is organized as follows. Section II introduces an overview of our proposed system. Section III provides the details of each module for "passin a ball" task. Section IV shows the experimental result of the task to use integrated modules. Finally concluding remarks are given.

II. Task, Robot, and Environment

A. Robot platforms

Fig. 1 shows biped robots used in the experiments, HOAP-1, HOAP-2, and their on-board views. HOAP-1 is 480 [mm] in height and about 6 [kg] in weight. It has a one-link torso, two four-link arms, and two six-link legs. The other robot, HOAP-2, is a successor of HOAP-1. It is 510 [mm] in height and about 7 [kg] in weight. It has two more joints in neck and one more joint at waist. Both robots have four force sensing registors (FSRs) in each foot

to detect reaction force from the floor and a CCD camera with a fish-eye lens or semi-fish-eye lens.

Fig. 1. HOAP-1 with fish-eye lens and HOAP-2 with semi-fish-eye lens

These robots detect objects in the environments by colors. In this experiment, a ball is colored orange, and the knees of the opponent robot are colored yellow. The centers of these colored regions in the images are recorded as the detected positions.

B. Visuo-motor learning

Let the motion flow vector $\Delta \boldsymbol{r}$ at the position $\boldsymbol{r}$ in the robot's view when a robot takes a motion, a. The relationships between them can be written,

$$\Delta \boldsymbol{r} = f(\boldsymbol{r}, a), \quad (1)$$

$$a = g(\boldsymbol{r}, \Delta \boldsymbol{r}). \quad (2)$$

The latter is useful to determine the motion parameters after planning the motion path way in the image. However, it is difficult to determine one motion to realize a certain motion flow because different motion primitives can produce the same image flow by adjusting motion parameters. So, we separate the description of the relationship between the motion parameters in each primitive and the image flow as follows.

$$\boldsymbol{a}^i = (p_1^i, \ldots, p_n^i)^T = g_p^i(\boldsymbol{r}, \Delta \boldsymbol{r}) \quad (3)$$

$$\Delta \boldsymbol{r} = f^i(\boldsymbol{r}, \boldsymbol{a}^i), \quad (4)$$

$\boldsymbol{a}^i$ is a motion parameter vector of the i-th motion primitive. We use neural networks to learn these relationships.

C. Task and Assumptions

"Face-to-face pass" can be decomposed into following three modules:

1) approaching to a ball to kick,
2) kicking a ball to the opponent, and
3) trapping a ball which is coming to the player

All these basic modules need the appropriate relationships between motion parameters and the environment changes. For example, to trap a ball appropriately, the robots must estimate the arrival time and the position of the coming ball. To approach to a kicking position, the robot should know the causal relationship between the walking parameters and the positional change of the objects in its image. Further, to kick a ball to the opponent, the robot must know the causal relationship between the kicking parameters and the direction the kicked ball will go.

Moreover, basic modules to realize these behaviors should be activated at the appropriate situations. Here, the designer determines these situations to switch the behaviors, and we focus on the module learning based on the optic flow information. Fig. 2 shows an overview of the proposed system.

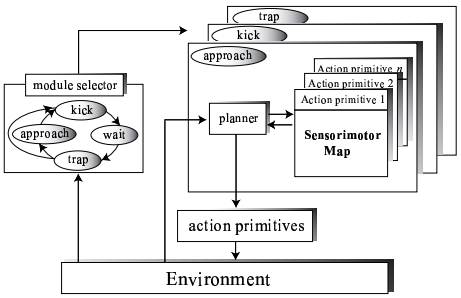


Fig. 2. A system overview

III. Module Learning Based on Optic Flow Information

A. Ball Approaching

Approaching to a ball is the most difficult task among the three modules because this task involves several motion primitives each of which has parameters to be determined. These motions yield various types of image flows depending on the values of the parameters which change continuously. We make use of environmental image flow patterns during various motions to approach to the ball.

We separate the description of the relationship between the motion and the image flow into the relationship between the motion primitive and the image flow, and the relationship between the motion parameters in each primitive and the image flow (Fig. 3), as follows:

$$m_i = g_m(\boldsymbol{r}, \Delta \boldsymbol{r}), \quad (5)$$

$$\boldsymbol{a}^i = (p_1^i, p_2^i)^T = g_p^i(\boldsymbol{r}, \Delta \boldsymbol{r}), \quad (6)$$

$$\Delta \boldsymbol{r} = f^i(\boldsymbol{r}, \boldsymbol{a}^i), \quad (7)$$

where m_i is an index of the i-th motion primitive and $\boldsymbol{a}^i = (p_1^i, p_2^i)^T$ is the motion parameter vector of the i-th motion primitive. In this study, the motion primitives related to this module consists of 6 primitives; *forward walk* (left and right), *curve walk* (left and right), and *side step* (left and right). Each of the primitives has two parameters which have real values, as shown in Fig. 4.

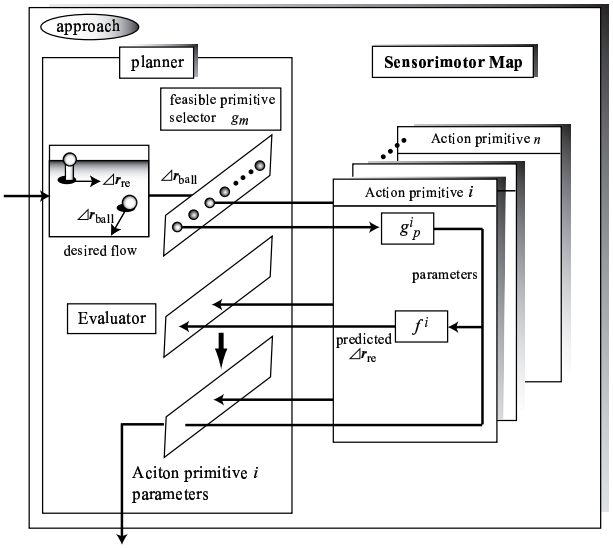


Fig. 3. An overview of the approaching module

Given the desired motion pathway in the robot's view, we can select appropriate primitive by g_m, and determine the motion parameters of the selected motion primitive by g_p^i based on the learned relationships among the primitives, their parameters, and flows. If the desired image flow yields several motion primitives, the preferred motion primitive is determined by an evaluation function.

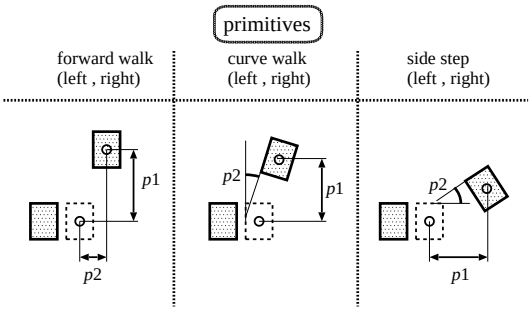


Fig. 4. Motion primitives and parameters for approaching

Images are recorded every step and the image flow is calculated by block matching between the current image and the previous one. The templates for calculating flows are 24 blocks in one image as shown in Fig. 5.

g_m: All of the data sets of the flow and its positional vector in the image, $(\boldsymbol{r}, \Delta\boldsymbol{r})$, are classified by the self organizing map (SOM), which consists of 225 (15×15) representational vectors. After organizing, the indices of motion primitives are attributed to each representational vector. Fig. 6 shows the classified image vector (the figure at the left side) and the distribution of each primitive in SOM. This SOM outputs the index of appropriate motion primitive so that the desired flow vector in the image is realized.

f^i, g_p^i: The forward and inverse functions that correlates the relationship between the motion parameters in each

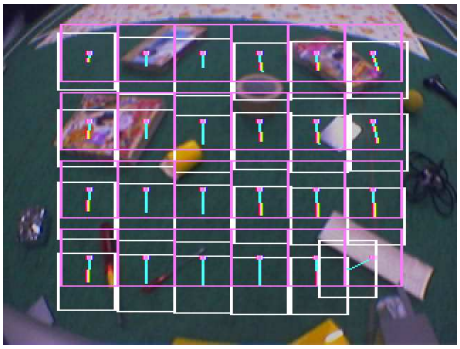


Fig. 5. An example of an optic flow in the robot's view

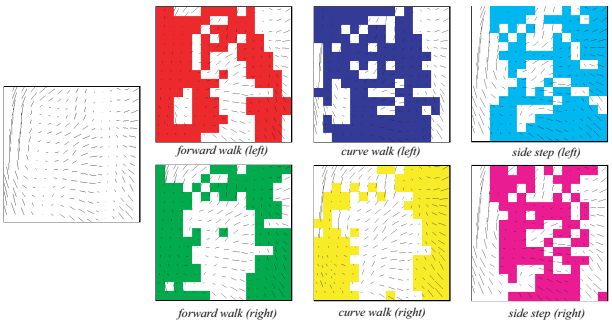


Fig. 6. Distribution of motion primitives on the SOM of optic flows

primitive and the image flow, f^i, g_p^i, are realized by a simple neural network. The neural network in each primitive is trained so that it outputs the motion parameters when the flow vector and the positional vector in the image are input.

plannning and evaluation function: In this study, the desired optic flows in the robot's view for the ball and the receiver, $\boldsymbol{s}_{ball}, \boldsymbol{s}_{re}$, are determined as vectors from the current position of a ball to the desired position (kicking position) in the robot's view, and as the horizontal vector from the current position to the vertical center line, respectively. The next desired optic flow of a ball to be realized, $\tilde{\boldsymbol{s}}_{ball}$, is calculated based on these desired optic flows,

$$n_{step} = \|\boldsymbol{s}_{ball}\|/\Delta r_{max}, \quad (8)$$

$$\tilde{\boldsymbol{s}}_{ball} = \boldsymbol{s}_{ball}/n_{step}, \quad (9)$$

where Δr_{max} is the maximum length of the experienced optic flow. This reference vector is input to the primitive selector, g_m, and the candidate primitives which can output the reference vector are activated. The motion parameters of the selected primitive are determined by the function g_p^i,

$$\boldsymbol{a}^i = g_p^i(\boldsymbol{r}_{ball}, \tilde{\boldsymbol{s}}_{ball}), \quad (10)$$

where $\boldsymbol{r}_{ball}$ is the current ball position in the robot's view. When the primitive selector outputs several candidates of primitives, the evaluation function depending on the task, $V(m_i)$, determines the preferred primitive. In this study, robots have to not only approach to a ball but also take an

appropriate position to kick a ball to the other. For that, we set the evaluation function as follows,

$$\begin{aligned} V(m_i) &= \|\tilde{\boldsymbol{s}}_{ball} - f^i(\boldsymbol{r}_{ball}, \boldsymbol{a}^i)\| \\ &\quad + k\|\mathbf{s}_{re} - n_{step} f^i(\boldsymbol{r}_{re}, \boldsymbol{a}^i)\|, \qquad (11) \\ P &= \underset{i \in primitives}{\arg\min} \; V(m_i) \end{aligned}$$

where k is the constant value, $\boldsymbol{r}_{re}$ is the current position of the receiver in the robot's view, and P is the selected primitive.

Fig. 7 shows experimental results of approaching to a ball. A robot successfully approach to a ball so that the hypothetical opponent (a poll) comes in front of it.

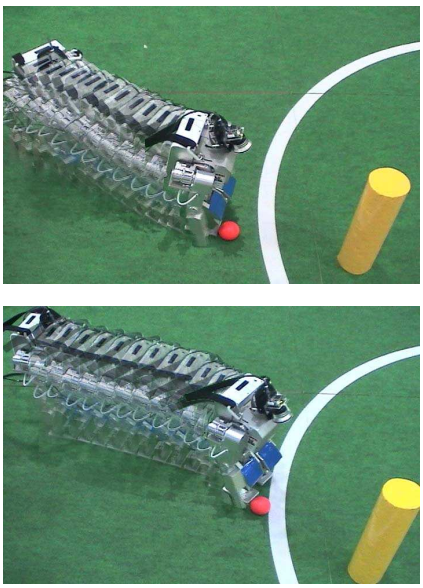


Fig. 7. Experimental results of approaching to a ball

B. Ball Kicking to the Opponent

It is necessary for the robots to kick a ball to the receiver very precisely because they cannot sidestep quickly. We correlate the parameter of kicking motion with the trace of the kicked ball in the robot's view so that they can kick to each other precisely. Fig. 8 shows a proposed controller for kicking.

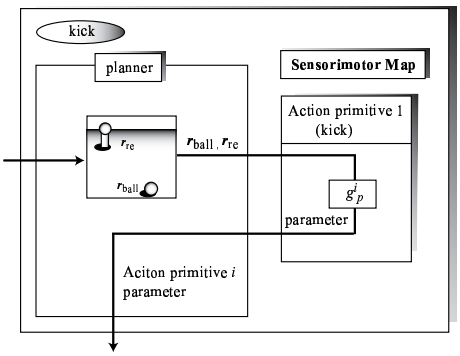
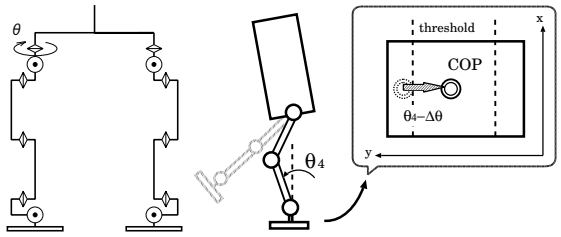


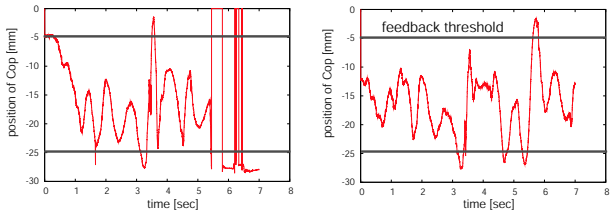
Fig. 8. The system for kicking module

The kicking parameter is the hip joint angle shown in Fig. 9 (a). A quick motion like kicking changes its dynamics depending on its motion parameter. The sensor feedback from the floor reaction force sensors is used for stabilizing the kicking motion. The displacement of the position of the center of pressure (CoP) in the support leg is used as feedback to the angle of the ankle joint of the support leg (see Fig. 9(b)). ,Fig. 9 (c) shows the effectiveness of the stabilization of the kicking motion.



(a) Kick parameter

(b) An overview of stabilization of kick motion



without feedback

with feedback

(c) The trajectories of CoP of the support leg during kicking motion

Fig. 9. The parameter and the stabilization of kicking

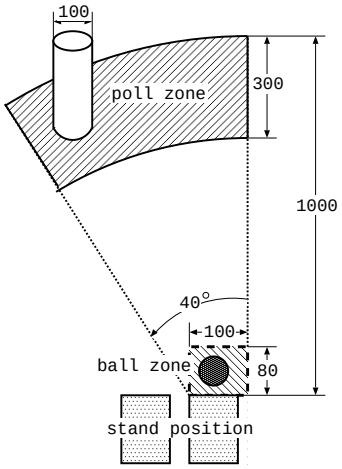
The initial ball position and the parameter of the kicking motion affects sensitively the ball trace in the robot's view. To describe the relationship among them, we use a neural network, which is trained in the environment where the poll (10 [cm]) is put about 1 [m] in front of the robot (Fig. 10 (a)). The trace of the ball (the effects of the self motion is subtracted) is recorded every 100 [msec], and the weights in the neural network are updated every one trial. Fig. 10 (b) shows the time course of error distance between target poll position and kicked ball in the robot's view. It shows that the error is reduced rapidly within 20 [pixel], which is the same size of the width of the target poll. Fig. 11 shows the kicking performance of the robot.

C. Ball Trapping

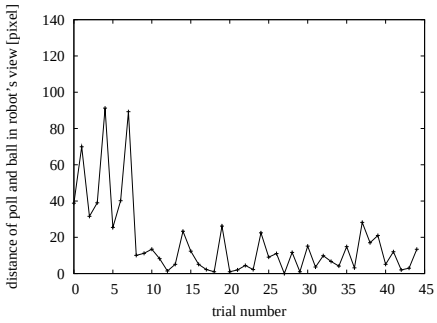
Fig. 12 shows an overview of trapping module. Robots learn the relationship between the position of the foot in robot's view and the trap parameter which affects the position of the foot, to acquire the skill to trap a coming ball.

Fig. 14 shows the trapping motion by HOAP-2 acquired by the method described below. In order to realized such a

Fig. 11. An experimental result of kicking a ball to the poll



(a) The environmental setting



(b) The error of learning kicking

Fig. 10. The environmental setting and the learning curve for kicking

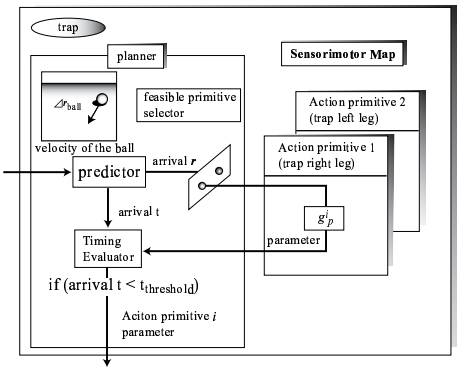


Fig. 12. An overview of trapping module

motion, the robot has to predict the position and the arrival time of a ball from its optic flow captured in the robot view. For that purpose, we use a neural network which learns the causal relationship between the position and an optic flow of the ball in visual image of a robot and the arrival position and time of the coming ball. This neural network is trained by the data in which a ball is thrown to a robot from the various positions. Fig. 13 shows several prediction results of the neural network after learning. Δx [pixel] and Δt [sec] indicates the errors of the arrival position and the time predicted at each point (every 0.3[sec]) in the robot's view. T denotes a duration of the ball rolling. Based on this neural network, the robots can activate the trapping motion primitive with the appropriate leg (right or left) at the appropriate timing (Fig. 14).

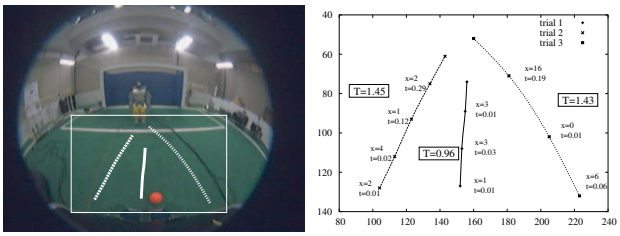


Fig. 13. The prediction of the position and time of a coming ball

IV. Integration of the Modules for Face-to-face Pass

To realize passing a ball between two humanoids, the basic modules described in the previous sections are integrated by a simple rule as shown in Fig. 15.

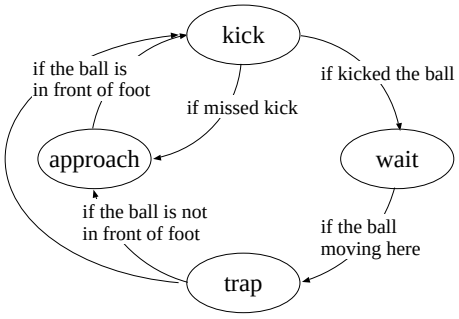


Fig. 15. The rule for integrating motion modules

Fig. 16 shows the experimental result. Two humanoids with different body and different camera lens realize the appropriate motions for passing a ball to each other based

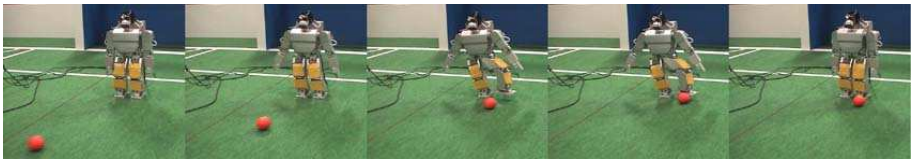


Fig. 14. An experimantal result of a trapping module

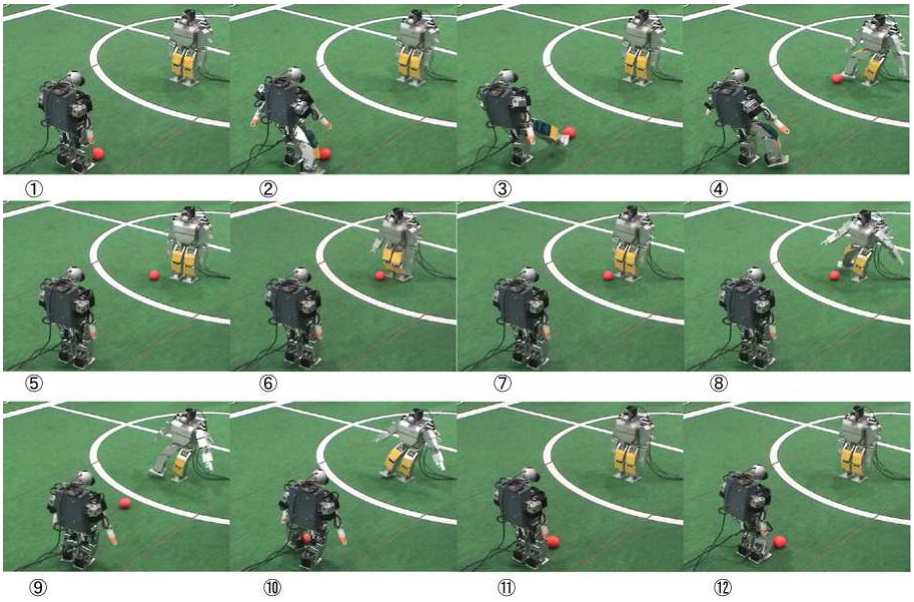


Fig. 16. An experimental result of passes between two humanoids

on their own sensorimotor mapping. The passing lasts more than 3 times.

V. Conclusions

In this paper, the robots learn the sensorimotor mapping between optic flow information and motion parameters. Acquiring basic modules for passing a ball is achieved using the sensorimotor mapping. In each module, optic flow information is correlated with the motion parameters. Through this correlation, a humanoid robot can obtain the sensorimotor mapping to realize the desired modules. The experimental results show that a simple neural network quickly learns and models well the relationship between optic flow information and motion parameters of each motion primitive. However, there remain the harder problems we skip in this paper. First is module decomposition problem, that is how to determine what are the basic modules for the given task. Second is planning, that is how to organize each motion primitive to achieve the given task. In this paper, we assume module decomposition and planning are given in advance. Combining the learning in each module level with that in higher level is our future issue.

References

[1] P. Fitzpatrick. First Contact: an Active Vision Approach to Segmentation, In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2161–2166, 2003.

[2] K. F. MacDorman, K. Tatani, Y. Miyazaki, M. Koeda and Y. Nakamura. Protosymbol emergence based on embodiment: Robot experiments, In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1968–1974, 2001.

[3] T. Nakamura and M. Asada. Motion Sketch: Acquisition of Visual Motion Guided Behaviors. In *Proc. of Int. Joint Conf. on Artificial Intelligence*, pp. 126–132, 1995.

[4] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba and H. Inoue, Walking Control System of a Humanoid for Tracking a Moving Object with Estimate of the Target Motion, In *Proc. of 3rd IEEE Int. Conf. on Humanoid Robots*, 2003.

[5] J. F. Seara, K. Strobl, and G. Schmidt. Information Management for Gaze Control in Vision Guided Biped Walking, In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 31–36, Lausanne, Switzerland, October 2002.

〒 162 東京都新宿区津久戸町 4-7 OS ビル 402 号室 03-5261-3401 Fax: 03-5261-3402

(本研究会についてのお問い合わせは下記にお願いします.)

AI チャレンジ研究会
主 査
奥乃 博
京都大学大学院 情報学研究科
知能情報学専攻 音声メディア分野
〒 606-8501 京都市左京区吉田本町
Tel: 075-753-5376 Fax: 075-753-5977
okuno@nue.org

担 当 幹 事
浅田 稔, 光永 法明
大阪大学大学院 工学研究科
知能・機能創成工学専攻 創発ロボット工学講座
〒 565-0871 大阪府吹田市山田丘 2-1
Tel: 06-6879-7349 Fax: 06-6879-7348
{asada,mitchy}@ams.eng.osaka-u.ac.jp

Executive Committee
Chair
Hiroshi G. Okuno
Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University
Yoshida-honmachi Sakyo-ku,
Kyoto, 606-8501, JAPAN

Secretary in Charge
Minoru Asada and
Noriaki Mitsunaga
Dept. of Adaptive Machine Systems
Graduate School of Engineering
Osaka University
2-1 Yamada-oka, Suita,
Osaka 565-0871, JAPAN

SIG-AI-Challenges web page; http://winnie.kuis.kyoto-u.ac.jp/SIG-Challenge/