Tetsunari Inamura
Iwaki Toshima
Hiroaki Tanie
Department of Mechano-Informatics
Graduate School of Information
Science and Technology
University of Tokyo, Japan

Yoshihiko Nakamura
Department of Mechano-Informatics
Graduate School of Information
Science and Technology
University of Tokyo, Japan
and
CREST program
Japan Science and Technology Agency
Japan

Embodied Symbol Emergence Based on Mimesis Theory

Abstract

"Mimesis" theory focused in the cognitive science field and "mirror neurons" found in the biology field show that the behavior generation process is not independent of the behavior cognition process. The generation and cognition processes have a close relationship with each other. During the behavioral imitation period, a human being does not practice simple joint coordinate transformation, but will acknowledge the parents' behavior. It understands the behavior after abstraction as symbols, and will generate its self-behavior. Focusing on these facts, we propose a new method which carries out the behavior cognition and behavior generation processes at the same time. We also propose a mathematical model based on hidden Markov models in order to integrate four abilities: (1) symbol emergence; (2) behavior recognition; (3) self-behavior generation; (4) acquiring the motion primitives. Finally, the feasibility of this method is shown through several experiments on a humanoid robot.

KEY WORDS—imitation learning, symbol emergence, hidden Markov models, dynamics abstraction, mirror neurons

1. Nomenclature

N	number of nodes
M	number of motion elements
T	length of observation sequence
R	number of motions in the database
$\boldsymbol{O}$	sequence of observations
$\boldsymbol{o}_t$	observation at time t
a_{ij}	probability of a transition from state i to j
c_{jm}	weight of mixture component m in state j
$\boldsymbol{\mu}_{jm}$	vector of means for the mixture component m of state j
$\boldsymbol{\Sigma}_{jm}$	covariance matrix for the mixture component m of state j

The International Journal of Robotics Research
Vol. 23, No. 4–5, April–May 2004, pp. 363-377,
DOI: 10.1177/0278364904042199

2. Introduction

Research of humanoid robots has a long history and has accumulated a substantial amount of literature. The focus of early efforts was mostly on the dynamics and control of bipedial walking motion. Although it has not yet reached the level of a complete solution, with liability and adaptability, the hardware technology has been established for building autonomous humanoids (Hirai et al. 1998; Nishiwaki et al. 2000; Kuroki et al. 2001).

Recently, human behavioral science and intelligence has become conspicuous as a real research issue of robotics. Although the motivation of the artificial intelligence originated there, the physical limitations have forced or justified researchers to carry on their research in a limited scope of complexity. It would be a major challenge of contemporary robotics to study robotic behaviors and intelligence in the full scale of complexity. This could then mutually share research outcomes and hypotheses with the human behavioral science and human intelligence.

The discovery of mirror neurons (Gallese and Goldman 1998) has been a notable topic of brain science concerning such issues. Mirror neurons have been found in the brains of primates and humans, which fire when the subject observes a specific behavior and also fire when the subject starts to act in the same manner. It also locates on Broka's area, which has a close relationship with language management. This fact suggests that the behavior recognition process and behavior generation process are combined as the same information processing scheme. This scheme is nothing but a core engine of the symbol manipulation ability. Indeed, in the "mimesis theory" of Donald (1991), he said that symbol manipulation and communicative ability are founded upon behavior imitation, which is integration of behavior recognition and generation. We believe that a paradigm can be proposed taking advantage of the mirror neurons, with considerations of the contention of Deacon (1997) that language and the brain had caused each other to evolve.

So far, many researchers have tackled with the issues between the imitation learning for humanoids and human intelligence (Schaal 1999; Matarić 2000). There are some suggestions that a module structure of basic motions is needed for the symbolization and representation of complex behavior, such as the work by Schaal (1999). In the approach of Kuniyoshi, Inaba, and Inoue (1994), robots can reproduce complex behaviors from observation of human demonstration with the abstraction and symbolization. However, it is difficult to apply this to general recognition and reproduction processes because of the lack of dynamics point of view, which means the robots have to memorize the whole flow of basic behavior. Moreover, the basic behavior modules need to be designed by a developer. Samejima et al. (2002) have proposed an imitation learning framework with symbolization modules. In this case, a premise has been set that the sequence of symbols is given from others by communication, thus a certain representation model for dynamics of the whole-body motion is needed.

In this paper, we propose a mathematical model that abstracts the whole-body motions as symbols, generates motion patterns from the symbols, and distinguishes motion patterns based on the symbols. In other words, it is a functional realization of the embodied symbol emergence framework, which is inspired by the mirror neurons and the mimesis theory. Therefore, we call the framework the "mimesis model". The purpose of the research is to propose a methodology of mathematical design for the mimesis mode.

One as an observer would view a motion pattern of the other as the performer; the observer acquires a symbol of the motion pattern. He recognizes similar motion patterns and even generates it by himself. The observer would then need to modify it from the performer's motion to the observer's one, according to his own body condition. The model is developed using hidden Markov models (HMMs). One issue is to identify appropriate motion primitives that enable both motion recognition and generation. This problem is to be solved using continuous HMMs (CHMMs). The second issue is how to generate the motion patterns as a time-series of the motion primitives, which is to be solved adopting discrete HMMs (DHMMs). The acquired models are to be modified according to the observer's body. This is the third issue and is to be discussed as a problem of database managements for HMMs.

First, we introduce the mathematical model of mimesis in Section 2. In Section 3, computational methods for symbol emergence, motion recognition and generation are explained. In Section 4, we discuss how to develop and design the motion primitive representation. The conclusion follows experimental results in Section 5.

3. Mimesis Model Recognizing Others' Motion and Generating Self-Motion

In this section, we explain the outline of mimesis models by showing the difference between usual imitation models.

In an imitation learning framework MOSAIC, which has been proposed by Samejima et al. (2002) and Samejima, Doya, and Kawato (2003), plural dynamics and inverse dynamics modules for the prediction and control of motion are implemented in order to imitate the motion of others. This framework is based on bi-directional theory suggested by Miyamoto and Kawato (1998). Both aim to imitate human behavior and symbolize the motion patterns as motion primitives. One of disadvantages of these methods is that the motion of others is always needed as a reference pattern, because it has no ability of description for dynamics of time-series motion primitives. On the contrary, we aim not only to imitate the motion of others but also to abstract the time-series motion patterns as symbol representation. This causes a situation in which no reference motion pattern is needed, i.e., more flexible for symbol emergence from behavior imitation.

Here, we propose an imitation framework which abstracts the dynamics of the motion as symbol representations, recognizes the motions of others, and generates self-motions from the symbol representations. The realization of the framework leads to the implementation of the mirror neuron from an engineering point of view.

3.1. Mimesis Model Based on Hidden Markov Models

The mimesis model consists of three parts, the perception part, the generation part, and the learning part, as shown in Figure 1. In the perception part, observed motion patterns are analyzed into basic motion primitives, and the dynamics in the sequence of the elements is abstracted as symbol representations.

In the generation part, a sequence of motion elements is decoded from a proto-symbol. However, the generated motion patterns would be inappropriate for real humanoids. For this issue, we introduce the learning part where motion

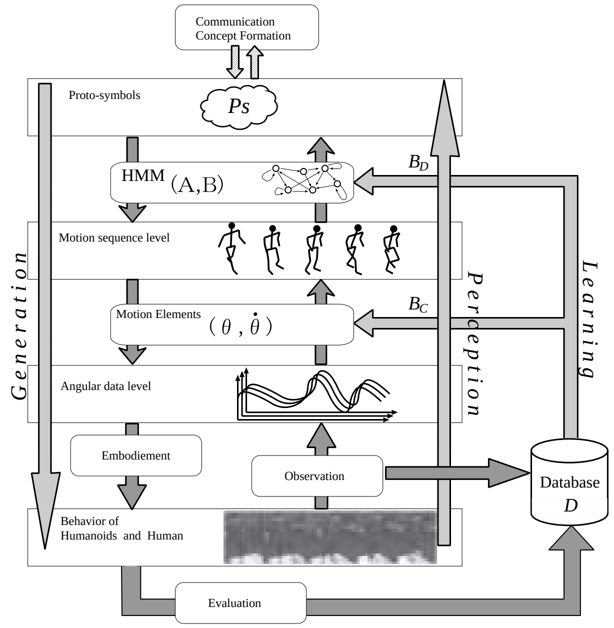

Fig. 1. An outline of the proposed mimesis model.

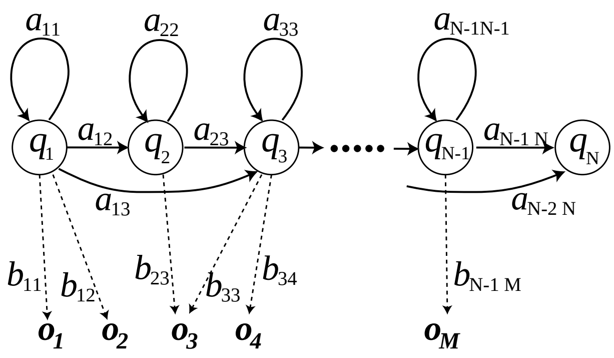

Fig. 2. A simple left-to-right type HMM.

elements are modified based on a database consist of performer's motions and observer's motions.

The characteristic needed by the mimesis model is to integrate three functions: motion recognition, motion generation and symbol emergence of motions. We focused on HMMs as the mathematical backbone for such integration. The HMM is a stochastic process that takes time-series data as an input, then outputs a probability that the data are generated by the model. The HMM is one of most famous tools as a recognition method for time-series data, especially in speech recognition. HMMs are divided into two types: DHMMs and CHMMs. The former treats sequences of discrete labels, and the latter treats sequences of continuous multi-dimensional vectors. In this subsection, we introduce the DHMM for the first step. The HMM consists of a finite set of states $\boldsymbol{Q} = \{q_1, \dots, q_N\}$, a finite set of output label $\boldsymbol{S} = \{o_1, \dots, o_M\}$, a state transition probability matrix $\boldsymbol{A} = \{a_{ij}\}$, an output probability matrix $\boldsymbol{B} = \{b_{ij}\}$, and an initial distribution vector $\boldsymbol{\pi} = \{\pi_i\}$, that is $\{\boldsymbol{Q}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. In this framework, state transition is performed probabilistically and some labels $\boldsymbol{o}_i$ are output during the transition, as shown in Figure 2.

3.2. Motion Elements

In order to connect discrete symbol representations and time-series motion data, a motion element is introduced. A motion element corresponds to a point in a phase space which consists of joint angle of humanoids, velocity, acceleration, and so on as follows:

$$\boldsymbol{u} \overset{\text{def}}{=} \boldsymbol{\mu}. \tag{1}$$

More generally, adopting the displacement and velocity of the base link and hands, and various sensory information, is effective for recognizing and generating more complex behavior. In this paper, we do not determine the type of physical quantity or concrete target behavior. We take the stance that a developer determines the task and physical quantity according to need.

A time-series motion pattern

$$\boldsymbol{O} = [\boldsymbol{o}_{k_1}\boldsymbol{o}_{k_2} \dots \boldsymbol{o}_{k_T}] \tag{2}$$

$$k_i \in \{1, 2, \dots, M\} \tag{3}$$

indicates motion of others and self-motion at the same time in the HMM, by correspondence of the M pieces of motion element $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_M)$ to output label o as follows:

$$\boldsymbol{o}_i = \boldsymbol{u}_i, \tag{4}$$

where $\boldsymbol{O}$ indicates a row vector which consists of a sequence of motion elements $\boldsymbol{o}$. The ith element from the left indicates the motion element at the ith discrete time. The question of what type of physical quantity is effective for the model is affected by the characteristic of target behavior. In this paper, we have adopted simple joint angle space as the motion element for the first step, and we propose motion recognition, generation and abstract method independent of the type of physical quantity.

3.3. HMMs as a Proto-Symbol

Definition by eq. (4) is nothing but a connection between a label (or an index) and feature vector for a certain moment. To represent the dynamics of feature vector sequence, certain

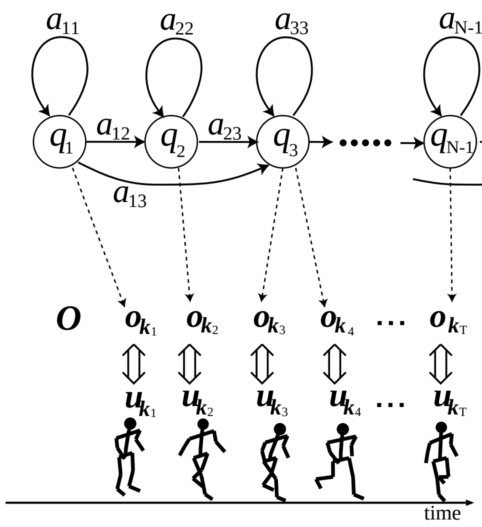
a_{11}
a_{22}
a_{33}
$a_{\text{N-1N-1}}$
q_1
a_{12}
q_2
a_{23}
q_3
$q_{\text{N-1}}$
$a_{\text{N-1 N}}$
q_{N}
a_{13}
$a_{\text{N-2 N}}$
$\boldsymbol{O}$
$\boldsymbol{o}_{k_1}$ $\boldsymbol{o}_{k_2}$ $\boldsymbol{o}_{k_3}$ $\boldsymbol{o}_{k_4}$ … $\boldsymbol{o}_{k_T}$
$\boldsymbol{u}_{k_1}$ $\boldsymbol{u}_{k_2}$ $\boldsymbol{u}_{k_3}$ $\boldsymbol{u}_{k_4}$ … $\boldsymbol{u}_{k_T}$
time

Fig. 3. Motion elements and HMMs.

representation methods are needed. Symbols may be defined in a narrow sense as ones with embodied meaning and their mutual distances. We propose to consider the acquired HMMs as symbols. Although, in the scope of this paper, they have embodied meanings, their mutual distances have not yet been introduced. We plan to discuss this in future work. In this sense, it would be appropriate to call the HMMs proto-symbols.

We shall concentrate on the HMM parameters. As left-to-right type HMMs as shown in Figure 2 are used in the mimesis model, the initial distribution vector $\boldsymbol{\pi}$ has a fixed value of $(1, 0, \ldots, 0)$. A set of states $\boldsymbol{Q}$ and a set of output labels $\boldsymbol{S}$ have no direct relationship between output time-series data. The state transitions probability matrix $\boldsymbol{A}$ and the output probability matrix $\boldsymbol{B}$ can be regarded as an abstraction parameter of probabilistic dynamics of the HMM.

Thus, we define the proto-symbols as follows:

$$\mathcal{P}_{\mathcal{S}} \overset{\text{def}}{=} \{\boldsymbol{A}, \boldsymbol{B}\}. \quad (5)$$

The HMM is a stochastic mathematical framework for sequential data. It is furnished with well-established algorithms of computation. The acquisition, recognition, and generation of motion patterns are to be efficiently computed using the algorithms. It is also known that HMMs are successfully used in speech recognition.

An alternative to HMMs for such computation is the use of recurrent neural networks (RNNs). RNNs also memorize dynamics of patterns (Morita 1996; Morita and Murakami 1997; Tani 2001; Inamura, Nakamura, and Simozaki 2002). The authors tested the use of RNNs for motion recognition and generation (Inamura, Nakamura, and Simozaki 2002). According to the result of Inamura, Nakamura, and Simozaki (2002), more than 500 nodes and more than 200,000 weight parameters between each node are needed in order to integrate the memorization and generation process on the same RNN. The RNN consists of motion element neurons, symbol representation neurons and buffer neurons for treating time-series data. The required number of weights increases in proportion to the square of the number of all nodes. In contrast, the number of parameters used in HMMs is proportional to the product of the number of nodes and motion elements. To give a concrete example, a HMM consists of 25 nodes and 80 motion elements and requires about 2500 parameters, in order to recognize and generate the motion. Therefore, the drawback of RNNs is in the low efficiency of computation; RNNs would use a large set of parameters to memorize a few motion patterns. The parameters would require a large computation to be adjusted.

4. Motion Abstraction, Recognition and Generation using HMMs

4.1. Creating Proto-Symbols Through Observation

Motion abstraction, i.e., proto-symbol generation, consists of two phases. In the first phase, observed motions are transferred into the sequence of the motion elements by a segmentation process. In the second phase, dynamics existing in the motion elements sequence is abstracted, and represented as proto-symbols.

In order to transform the observed motion pattern $\Theta(t)$ into a sequence of motion elements $\boldsymbol{O} = [\boldsymbol{o}_{k_1}\boldsymbol{o}_{k_2}\ldots\boldsymbol{o}_{k_T}]$, $\boldsymbol{\theta}$ for each short time period is sampled, then we calculate

$$j = \arg\max_i \frac{\exp\{-(1/2)(\boldsymbol{\theta}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_i)\}}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}_i}}. \quad (6)$$

The meaning of the above equation is letting j be i which causes the maximum value of the right side. D is the number of dimensions of the motion elements, det indicates the determinant of a matrix, and the superscript "T" indicates the transpose of a matrix. The right-hand side represents a Gaussian distribution function with a covariance matrix $\boldsymbol{\Sigma}$ and a mean vector $\boldsymbol{\mu}$. The calculation contributes to selecting a suitable motion element $\boldsymbol{u}_j$ which locates near to the sampled motion $\boldsymbol{x}$ in the phase space. Let $\boldsymbol{u}_j$ be a motion element for each short time period. $[\boldsymbol{u}_{k_1}\boldsymbol{u}_{k_2}\ldots\boldsymbol{u}_{k_T}]$, namely a sequence of motion elements, is output by a repetition of the above calculation for all short time periods.

After this, a parameter of a HMM ($\{\boldsymbol{A}, \boldsymbol{B}\}$), which outputs the sequential elements plausibly, is calculated and registered as a proto-symbol $\mathcal{P}_{\mathcal{S}}$. Humanoids gather several motion patterns as a stock of observed data for the learning. When an unknown motion is input, the robot creates a new HMM. $\boldsymbol{A}$ and $\boldsymbol{B}$ can be calculated by the Baum–Welch algorithm, which is one of the Expectation Maximization (EM) algorithms (Young et al. 2000).

4.2. Motion Recognition Using Proto-Symbols

To recognize motion of others, observed motions are transformed into a sequence of motion elements $\boldsymbol{O} = [\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t]$, and a parameter $P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}})$ is calculated. This parameter indicates a probability that a motion pattern $\boldsymbol{O}$ is generated by a proto-symbol $\mathcal{P}_{\mathcal{S}}$. This value is called a likelihood, calculated by the forward algorithm (Young et al. 2000).

Each proto-symbol corresponds to each motion, thus likelihood values of the input motion against each proto-symbol are calculated. A proto-symbol that corresponds to an input pattern should indicate high likelihood, and other proto-symbols ought to indicate low likelihood. In order to distinguish these likelihoods, the following criterion is introduced

$$R(\boldsymbol{O}) = \log \frac{\max\{P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}_i})\}}{\mathrm{second}\{P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}_i})\}}, \tag{7}$$

where second$(\boldsymbol{x})$ denotes the second highest value in the components of $\boldsymbol{x}$. The mimesis model recognizes the input motion without any confusion when R indicates a high value. In this case, the recognition result becomes $\mathcal{P}_{\mathcal{S}_j}$, where

$$j = \arg\max_i \{P(\boldsymbol{O}|\mathcal{P}_{\mathcal{S}_i})\}. \tag{8}$$

When R indicates a low value, the recognition fails and the mimesis model tries to shift to the proto-symbol creation phase.

4.3. Motion Generation Using Proto-Symbols

Basically, original patterns are decoded using the expectation operator in the stochastic model; however, applying the expectation operator is difficult in the HMM. The HMM has a two-stage stochastic process: state transition and label output. Applying the expectation operator is simple for the latter process, but difficult for the former process. The results of the recurrent state transition would not fit on the same dimensional phase space. For example, the length of a state sequence changes every trial. This means that integration of the probability values could not execute holomorphically. Therefore, we adopt the averaging method over repetition of motion generation. The detailed order of the generation is as follows.

1. Initialization. Let the starting node be q_1, let the node token be $i = 1$, and let the motion elements sequence be $\boldsymbol{O} = \phi$.
2. Deciding the transition destination node q_j using transition matrix $\boldsymbol{A}$ stochastically.
3. Deciding the output label $\boldsymbol{o}_{k_t}$ during the transition from node q_i to q_j stochastically using output matrix $\boldsymbol{B}$.
4. Adding the output label $\boldsymbol{o}_{k_t}$ to the motion elements sequence $\boldsymbol{O}$. $\boldsymbol{O} := [\boldsymbol{O}\ \ \boldsymbol{o}_{k_t}]$.
5. Let the generation process be stopped when the token reaches the end node q_N, or returns to step 2 letting $i := j, t := t + 1$.
6. Finally, the sequential motion elements are transformed into continuous joint angle representations.

The output motions using the above operations are not the same, but have different time lengths and orders of motion elements, because the output operations are stochastic. However, it is possible to generate an approximate motion pattern because the parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ represent the abstraction of dynamics in the motion pattern. Therefore, the above operations are repeated, and plural generated motions are averaged. As the time lengths of each generated motion are different, we make the time length uniform using

$$\theta'(t) = \theta\left(T \frac{t}{T_u}\right) \tag{9}$$

where T is the time length of each motion, and T_u is the time length of the uniformed motion. After this, each joint angle is averaged.

Several researches have already proposed motion recognition methods based on the HMM (Yamato, Ohya, and Ishii 1992; Pook and Ballard 1993; Wada and Matsuyama 1998; Yoshiike et al. 1998; Ogawara et al. 2002), however, no research exists in which motion is generated from the HMM. Kobayashi et al. (1996) and Imai, Tokuda, and Kobayashi (1995) have proposed a speech parameter generation method using the HMM; however, the generation process is not the opposite direction of the speech recognition process. The most important characteristic of our method is that the motion recognition and motion generation process are integrated by only a single HMM.

5. Development of Motion Elements Through Repetition of Motion Observation and Generation

The performance of motion recognition and generation is influenced by the characteristic of motion elements. If the number of elements were too few, the generation would fail. If the motion elements had no relationship between the observed motions, the recognition process would fail. Therefore, we have adopted an approach that the system searches the best motion elements with an evaluation criterion whether the generated motion would be fit for the body and the recognition would be succeeded against familiar motion. Using the method, the humanoid can acquire adequate motion elements through repetition of motion perception and generation.

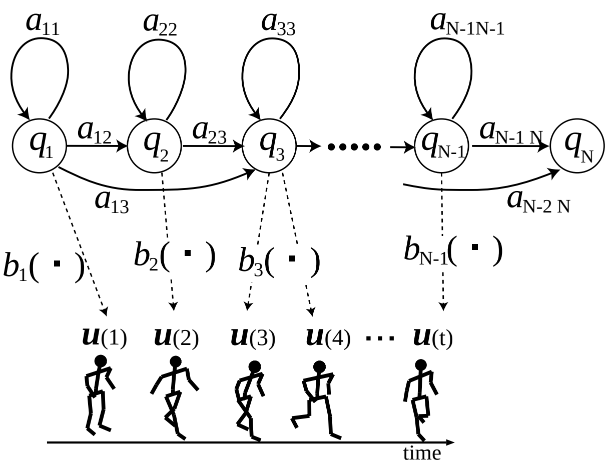

Fig. 4. A continuous hidden Markov model.

5.1. Introduction of Continuous Hidden Markov Models and Application to Mimesis Model

In this phase, we introduce CHMMs (Young et al. 2000), which can treat continuous multi-dimensional data. The difference between normal DHMMs and CHMMs is that the transition process outputs continuous multi-dimensional vectors, different from the DHMMs in which the discrete labels are output, as shown in Figure 4. In the CHMMs, the output probability matrix $\boldsymbol{B}$ becomes a probability density function. Here, the density function is approximated with linear combination of Gaussian functions as follows

$$P_i(\boldsymbol{o}) = \sum_{j=1}^{m} c_{ij} \mathcal{N}_{ij}(\boldsymbol{o}; \boldsymbol{\Sigma}, \boldsymbol{\mu}), \tag{10}$$

where $P_i(\boldsymbol{o})$ is the probability density function for the output of continuous vector $\boldsymbol{o}$ at the ith state node, m is the number of mixture Gaussian functions, and c_{ij} is the mixture coefficient. $\mathcal{N}(\boldsymbol{o}; \boldsymbol{\Sigma}, \boldsymbol{\mu})$ is the Gaussian function

$$\mathcal{N}_{ij}(\boldsymbol{o}; \boldsymbol{\Sigma}, \boldsymbol{\mu}) = \frac{\exp\left\{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu}_{ij})^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\theta}-\boldsymbol{\mu}_{ij})\right\}}{\sqrt{(2\pi)^D \det \boldsymbol{\Sigma}_{ij}}} \tag{11}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix, $\boldsymbol{\mu}$ is the mean vector, and D is the number of dimensions of the continuous vector $\boldsymbol{o}$.

The characteristics of the CHMMs are decided by the parameters $\{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{c}, \boldsymbol{\Sigma}, \boldsymbol{\mu}\}$. These parameters are calculated using the Baum–Welch algorithm.

Here, each mean vector of the Gaussian function is regarded as an important representation of the observed motion. Therefore, we divide the parameters of CHMMs, and redefine the motion elements $\boldsymbol{e}$ as follows:

$$\boldsymbol{u}_i \overset{\mathrm{def}}{=} \{\boldsymbol{\Sigma}_i, \boldsymbol{\mu}_i\}. \tag{12}$$

In other words, the number of motion elements is as many as the number of mixture Gaussian components. An important issue is that the motion elements are automatically calculated by the Baum–Welch algorithm, as mentioned in Section 4.1.

Motion elements could be regarded as the filter between continuous motion representation and discrete motion representation. When continuous motion is transferred into discrete motion, $\boldsymbol{e}_i$, where $i = \arg\max_j \mathcal{N}_j(\boldsymbol{o})$, is adopted as a typical motion element for each time period. When discrete motion is transferred into continuous motion, the sequence of $\boldsymbol{\mu}_i$ is used directly.

To sum up, the mimesis system have the following advantages with CHMMs.

- Motion elements are able to express the whole-body motion, therefore various motion patterns are available easily.
- The parameters of motion elements are automatically calculated.

5.2. Hybrid Hidden Markov Model

Although many advantages are available, CHMMs have a disadvantage that huge computational quantity is needed. It should take much time for motion generation and recognition. Therefore, we propose a hybrid HMM, which consists of CHMMs and DHMMs as shown in Figure 5.

In the motion recognition and generation phase, DHMMs are used in which the computational quantity is small. In the motion element acquisition phase, CHMMs are used in which the computational quantity is large.

5.3. Closing the Mimesis Loop for Embodiment

The parameters which decide the characteristics of HMMs and motion elements are acquired using the Baum–Welch algorithm (Young et al. 2000) which is a type of EM algorithm. This algorithm can be expressed by the following equations

$$\mathcal{D} = \{\boldsymbol{O}^1, \boldsymbol{O}^2, \ldots, \boldsymbol{O}^l\} \tag{13}$$

$$\{\boldsymbol{A}, \boldsymbol{B}\} := \mathcal{B}_D(\mathcal{D}) \tag{14}$$

$$\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\} := \mathcal{B}_C(\mathcal{D}), \tag{15}$$

where $\mathcal{B}_D$, $\mathcal{B}_C$ are operations using the Baum–Welch algorithm, and $\mathcal{D}$ is a database consisting of l observations. The initial database $\mathcal{D}^0$ consists of only the observed motions of others; that is, motion elements and proto-symbols which have no relationship between the learner's physical characteristic are acquired by the above operations. Therefore, let the proto-symbols and motion elements be acquired with database manipulation during repetitions of motion recognition and generation as follows.

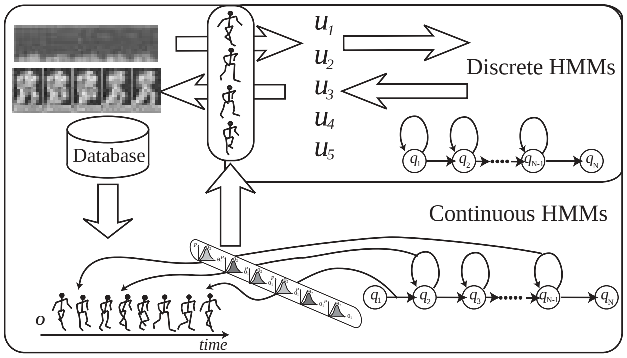

Fig. 5. A model of a hybrid hidden Markov model.

1. Generating a motion $\boldsymbol{O}$ from a proto-symbol $\mathcal{P}_{\mathcal{S}}$ and motion elements.

2. Judging whether the generated motion $\boldsymbol{O}$ is suitable or not.

3. Adding the motion to the database when the judged result is good: $\mathcal{D}^{t+1} := \mathcal{D}^{t} \cup \boldsymbol{O}$.

4. Acquiring the proto-symbols and motion elements using the above eqs. (14) and (15), and return to step 1.

For evaluation at step 2, two evaluation criteria were introduced: an inner evaluation for checking the characteristic of the proto-symbol, and an outside evaluation for checking the aim and meaning of the motion from the point of view of the teacher. For the outside evaluation, we prepared the following criterion

$$E_\theta = \frac{1}{T}\int_0^{\mathrm{T}} |\boldsymbol{\theta}_{in}(t) - \boldsymbol{\theta}_{out}(t)|\, \mathrm{d}t, \tag{16}$$

where $\boldsymbol{\theta}_{in}(t)$ and $\boldsymbol{\theta}_{out}(t)$ indicate the joint angles of an observed ideal motion and a generated motion, respectively. For the inner evaluation, the recognition rate $R(\boldsymbol{O})$ explained in Section 4.2 is used.

Considering the above two criteria, the following integrated criterion is used for the experiment

$$V = \alpha E_\theta + \beta R^{-1}(\boldsymbol{O}), \tag{17}$$

where α and β are certain constants. When the value V is larger than a certain threshold, the mimesis model judges that the ith motion data are suitable for the recognition process, it adds the motion data into the database, and calculates the motion elements again. These constants and the threshold are adjusted according to each experiment case.

At step 3, it is desirable that the generated self-motion and observed motions of others are distinguished. As the motion elements are used for both motion recognition and generation, a simple distinction leads to a deterioration of the process. Thus, a distinction strategy has been introduced that generated self-motions are stored in the database vividly, and the observed motions of others are stored dimly. Because of the distinction method, the influence of the initial motion of others would be decreases, and the database would be gradually under the control of generated self-motions. Actually, vivid motions are stored with little variance and dim motions are stored with large variance. In the learning phase, the number of motion samples in the database is controlled using the variance value.

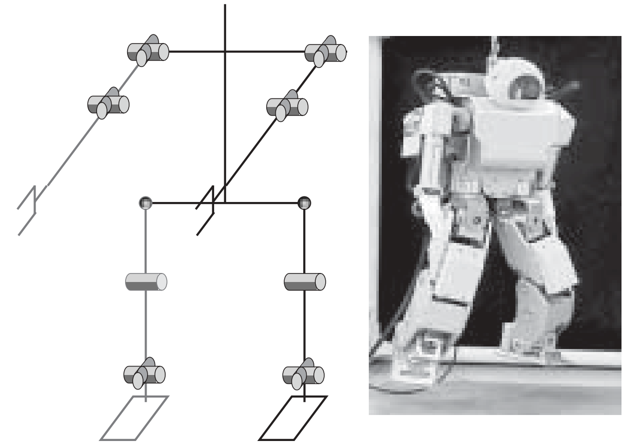

Fig. 6. The humanoid HOAP-1.

6. Experiments of Motion Element Acquisition

The humanoid used in the experiments is shown in Figure 6. The humanoid has four degrees of freedom at each arm, six degrees of freedom at each leg, namely 20 degrees of freedom for the whole body. We have confirmed the performance of our method by experiments where the mimesis model observes human motion and generates motions for a real humanoid. Using the Behavior Capturing System (Kurihara et al. 2002), joint angle data for 20 degrees of freedom are directly observed because the degrees of freedom of the humanoid are 20. The time period of each motion is about 2 s with a sampling time of 20 ms.

6.1. Experiments of Motion Generation

The humanoid used in this experiment has 20 degrees of freedom. We investigated the basic performance for squat behaviors. In the squat behavior, characteristic motion collected around the lower body. Therefore, we adopted a simple motion element which consists of three joint angles: hip (pitch

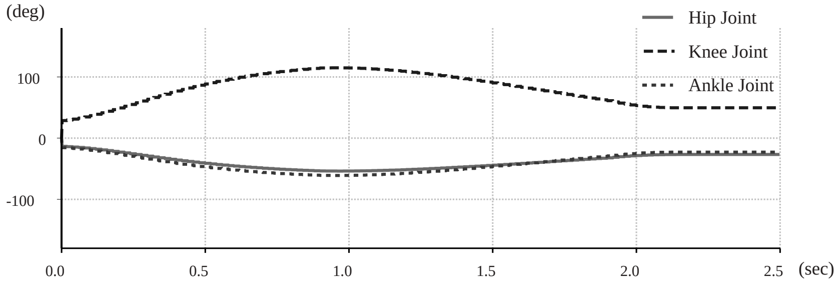

Fig. 7. Original motion pattern.

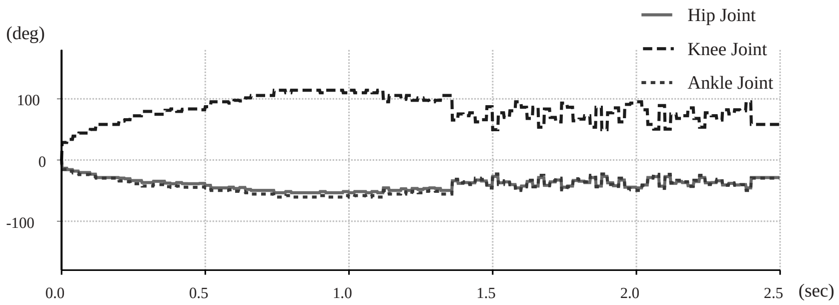

Fig. 8. A notion pattern using only one time generation.

axis), knee and ankle (pitch axis). In this subsection, we report on several experiments performed using the simple motion element.

Figure 8 shows the output motion pattern by one time output operation. Compared with an original motion pattern used in the learning (Figure 7), an approximate pattern is generated but the noise increases drastically. The cause of the noise is that the discrete motion elements are selected at each moment stochastically, thus the roughness and the discontinuity stood out.

Figure 9 shows the output motion pattern after 1000 times operation, as explained in Section 4.3. A CG animation using the pattern is shown in Figure 14. Compared with the motion pattern by one time operation (Figure 8), the joint angle became smoothed. There were some joint angle errors between the original patterns. We think that the cause of the error is the influence of coarse discrete motion elements.

The computational time for the generation process was about 1 s using a Pentium-III 1 GHz processor. The time is enough fast as the off-line pattern generator for humanoids. For this sort of problem, Okada, Tatani, and Nakamura (2002) have proposed a compression method in which a motion pattern of humanoids that have over 20 degrees of freedom is transferred into a three-dimensional vector. We think that reduction of the computational cost can be performed by adopting this method.

6.2. Experiments of Motion Recognition

For the motion recognition experiments, seven behaviors are prepared, as shown in Figure 10: (a) tennis swing (swing); (b) walking (walk); (c) Cossack dancing (dance); (d) kicking (kick); (e) backward walking (back); (f) crawling (crawl). The behaviors of (a)–(e) were treated as already-known motions, and the behavior (g) was treated as an unknown motion. Table 1 shows the recognition results.

The values in the table indicate the logarithm of likelihood $P(\boldsymbol{O}|\boldsymbol{A}, \boldsymbol{B})$. Proto-symbols arranged lengthways indicate the target of the recognition, and behavior names arranged sideways indicate the proto-symbols already learnt. The value indicates larger, the target motion matches better with the proto-symbol. The value of a certain target motion against a proto-symbol which corresponds to the motion indicates high, i.e., located on a diagonal line. The values of unfamiliar motion (unknown) against each proto-symbol are almost the same. Thus, we see that the recognition process would succeed without mistake, when the recognition rate R is set to about 1000 empirically.

6.3. Experiments of Motion Element Acquisition

For this experiment, four types of motion were recorded: walking, squating, picking up, and Cossack dancing. As in Section 6.1, there are three dimensions of the motion elements: hip joint (pitch), knee and ankle joint (pitch).

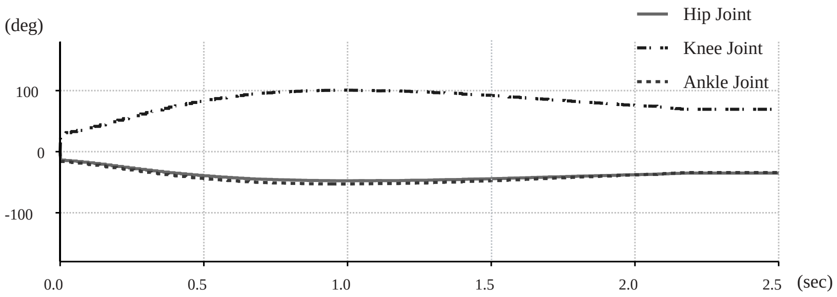

Fig. 9. A motion pattern using 1000 times generation.

Table 1. Recognition Results of the Motion of Others Using HMMs

	Proto-symbols						
Input Behavior	Swinging	Walking	Dancing	Kicking	Backward Walking	Crawling	R
Swing	−430	−3915	−4077	−3940	−4114	−4007	3485
Walking	−3048	−225	−3071	−1646	−3099	−3019	1420
Dance	−1656	−1603	−144	−1613	−1683	−1577	1433
Kicking	−2543	−1574	−2562	−199	−2585	−2519	1374
Backward walking	−2395	−2318	−2413	−2332	−202	−2372	2117
Crawling	−4083	−3950	−3815	−3976	−4151	−488	3327
Unknown behavior	−1915	−1853	−1928	−1865	−1946	−1896	11

After the 50 observations, the motion generation process is executed 50 times, and appropriate motions are added into the database. Figure 11 shows the acquired motion elements. Dots indicate the motion element, and the solid lines indicate the original motion's trajectory. As the figure shows, the motion elements are located near the original motion; that is, our method shows good performance.

6.4. Experiments of Motion Element Development Based on Embodiment

Here, we set up a situation where the joint angle limitation of the humanoid's knee is about 40 deg, less than the human one. We investigated whether motion elements for the humanoid are acquired by observations of human motions under such a condition. In the experiment, an 80 times loop is repeated as explained in Section 5.3.

Figure 12 shows the original motion which is performed by a human. Figure 15 shows the acquired motion elements from the performance when the joint angle limitation does not exist. A result with the limitation condition is shown in Figure 16. In these figures, three axes indicate hip joint (pitch), knee joint and ankle joint (pitch), as in Section 6.1. The curved line in the figures corresponds to the motion trajectory. The dots indicate acquired motion elements. As Figure 15 indicates, the motion elements are located near the original motion trajectory. Compared with Figure 15, motion elements are gathered not only on the A area, but also on the B area in Figure 16. These motion elements located on the B area are acquired by the generated self-motions in the database, which fits for the humanoid embodiment. This result shows that both motion elements are acquired: elements for the recognition of motions of others (A area) and those for the generation of self-motion (B area).

6.5. Designing Hidden Markov Models

Here, we concentrate on the rest parameter, namely the structure of HMMs. As the HMMs adopted in this paper are left-to-right type, the rest parameter is the number of nodes. It is possible to use the evaluation criterion explained in Section 5.3 for investigating the number of nodes, during the repetition of motion recognition and generation.

A tennis swing is selected for the experiment. The error value E_θ is measured by changing the number of nodes from 10 to 40. The result is shown in Figure 18. As the diagram indicates, the error value decreases hardly where the number of nodes shifts from 24 to 25. Figure 17 shows the generated

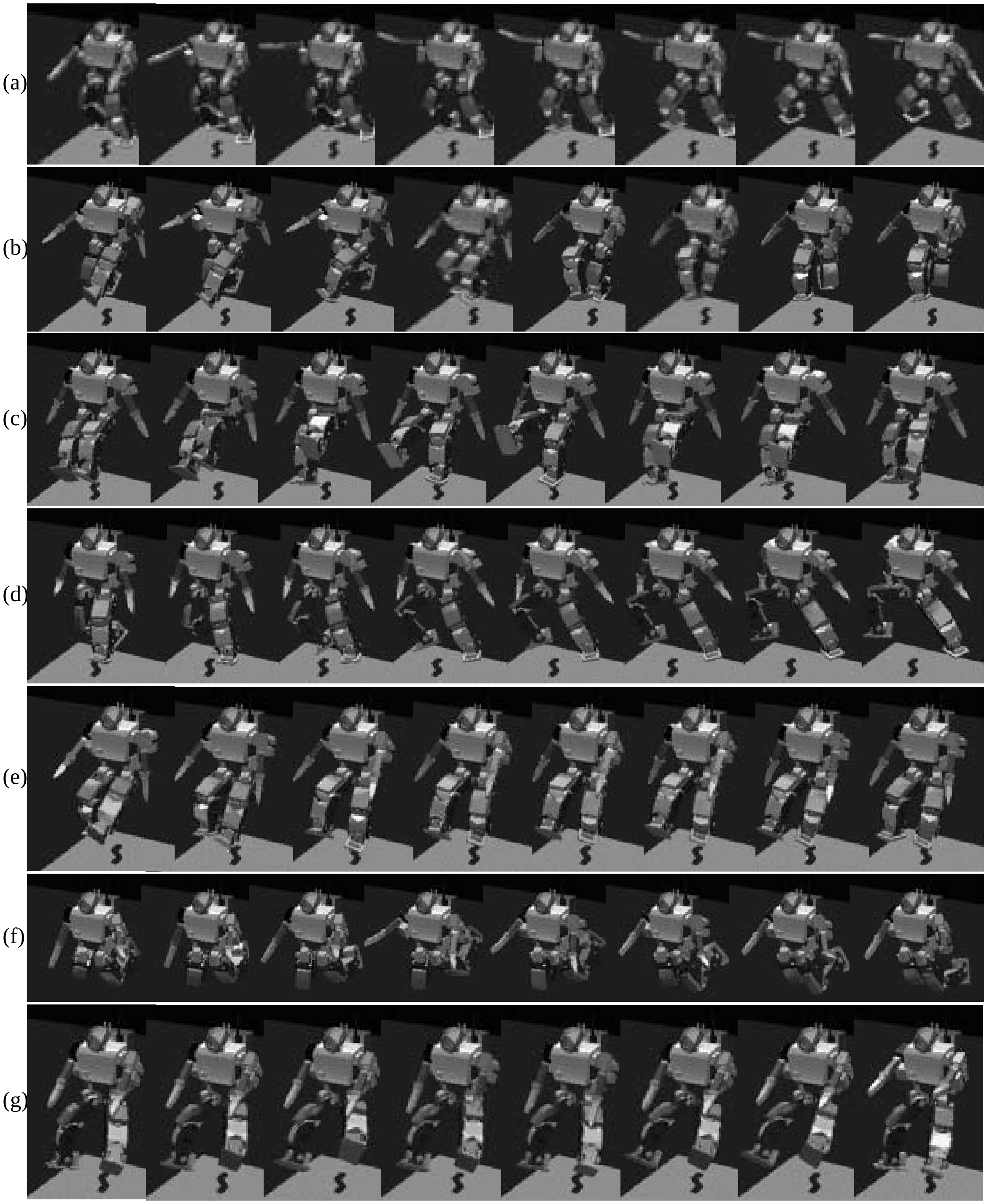

Fig. 10. Target behaviors: (a) tennis swing; (b) walking; (c) Cossack dance; (d) kicking; (e) backward walking; (f) crawling; (g) unknown behavior.

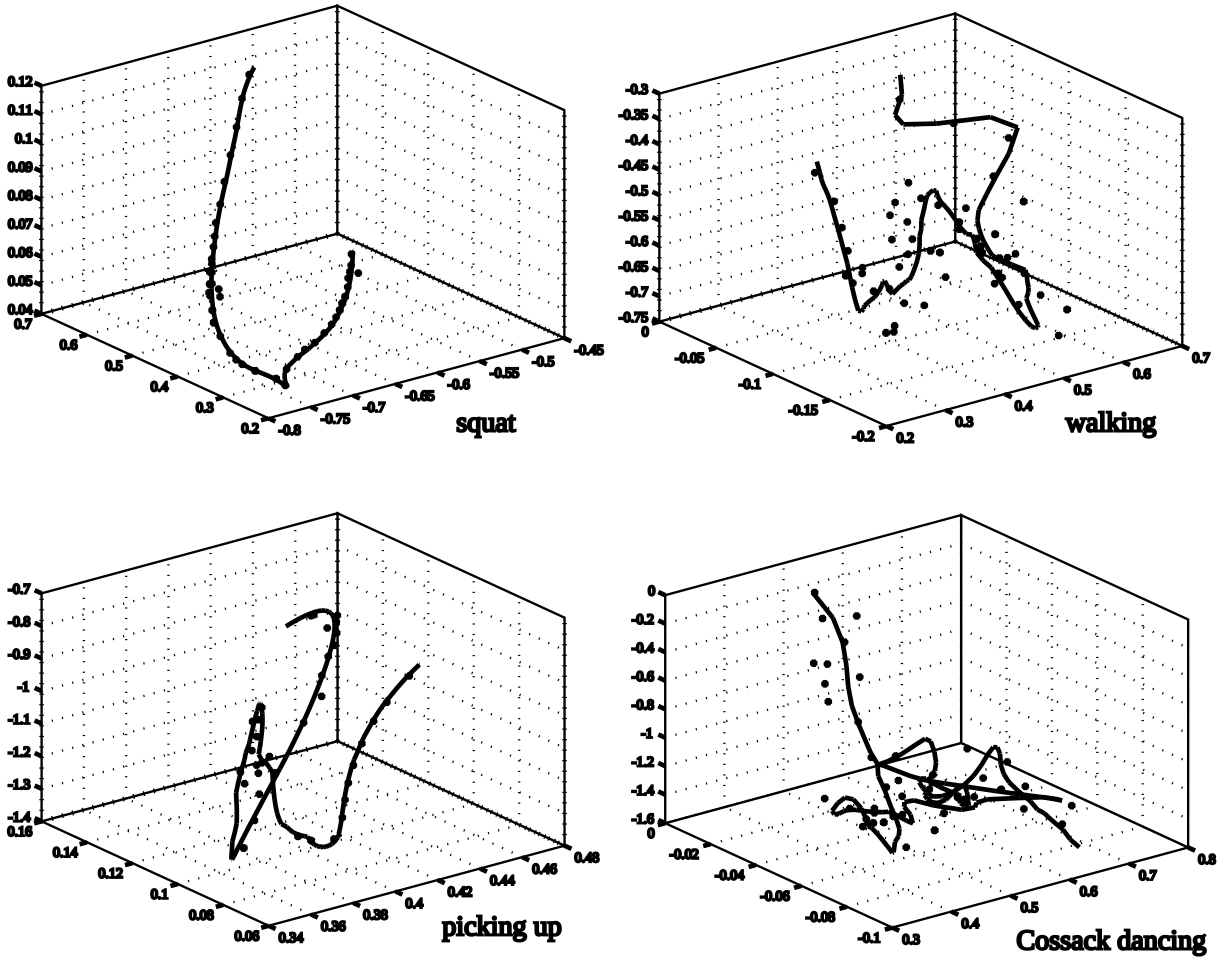

Fig. 11. A result of motion element acquisition against four types of motion.

Fig. 12. Motion capture system: step motion for learning data.

motion pattern for four conditions; the number of nodes is 20, 24, 25, and 40, respectively. The diagram focuses on the right shoulder's yaw joint. Solid lines indicate generated motion pattern, and dashed lines indicate the original motion pattern. The diagram supports the result that the desirable number of nodes is above 25.

7. Conclusions

In this paper, we have proposed a framework called the "mimesis model", which integrates motion recognition/generation and symbolization of motion patterns based on mimesis theory. In our mimesis model, proto-symbols and

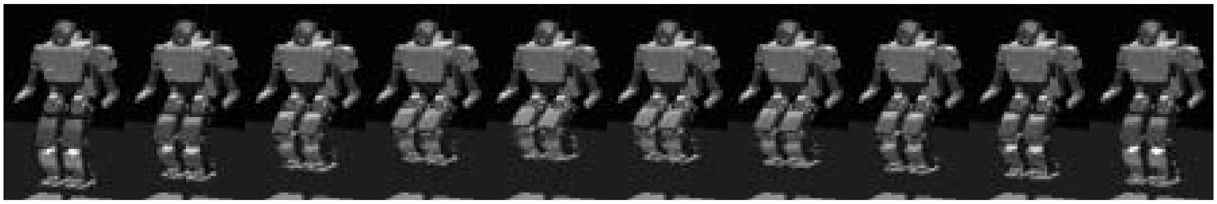

Fig. 13. Original motion for proto-symbol creation.

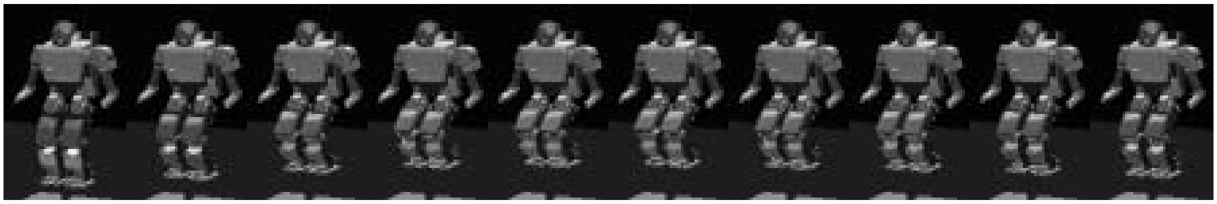

Fig. 14. Generated motion from proto-symbol.

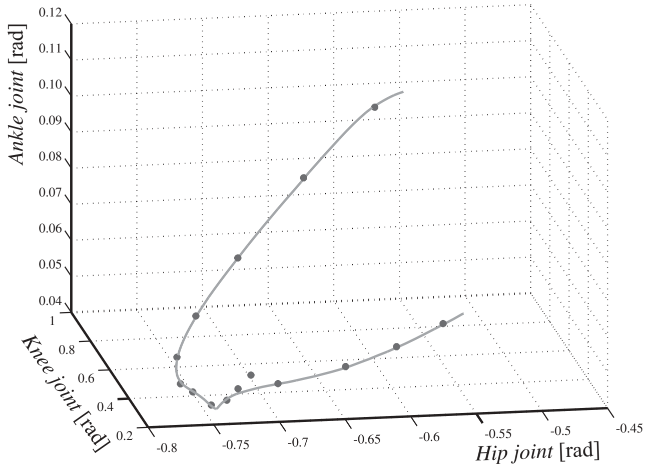

Fig. 15. Acquired motion elements without loop structure.

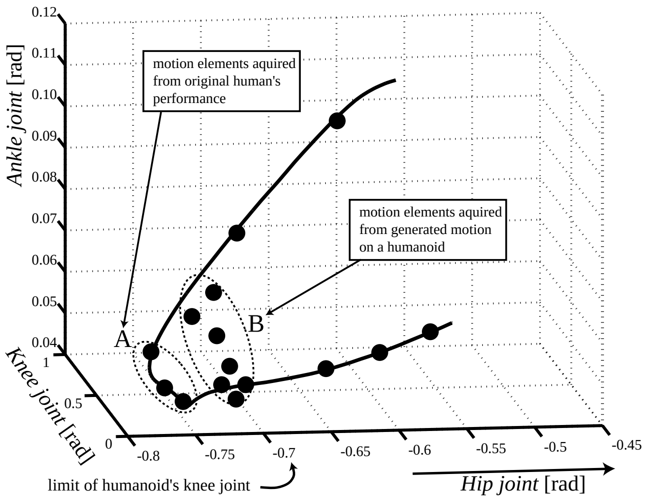

Fig. 16. Acquired embodied self-motion elements using loop structure.

motion elements are introduced with HMMs in order to integrate the following three abilities using only one mathematical model: (1) abstraction of motion patterns and symbol representation; (2) generation of self-motions from the symbol representation; (3) recognition of motions of others using the symbol representation. Through experiences, the feasibility of the mimesis model is clarified. Furthermore, we proposed an approach in which the development of motion elements is a result of the management of the motion database. We investigated the effectiveness through an experiment in which the learner's physical body condition is different from that of the teacher.

The mimesis model is not a simple method for motion recognition, generation, and abstraction. The recognition process, which transfers an observed motion of others into proto-symbol representation, and the generation process, which transfers a proto-symbol representation into self-motions, are implemented as opposite direction functions by only one mathematical model. The most important characteristic, integration between imitation learning and symbol emergence, is established by defining the bidirectional computation model as proto-symbols.

At the current stage, the proposed model can be applied to simple motion patterns; however, the application of the method to complex behavior is difficult, because consideration of the external environment is needed, such as tracking

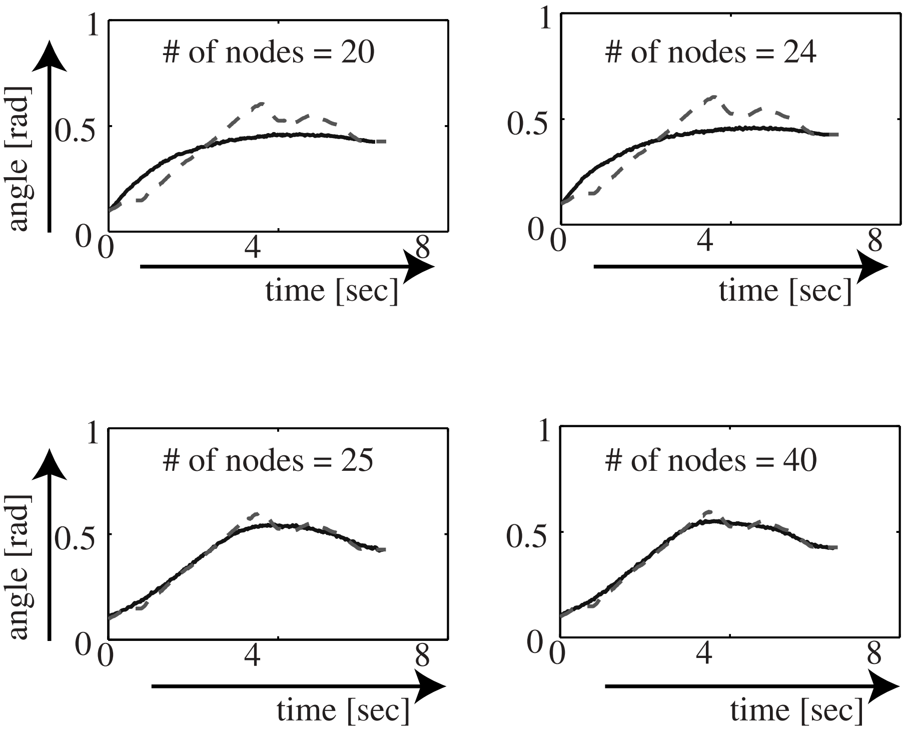

Fig. 17. Generated motion (shoulder yaw joint) for each number of nodes.

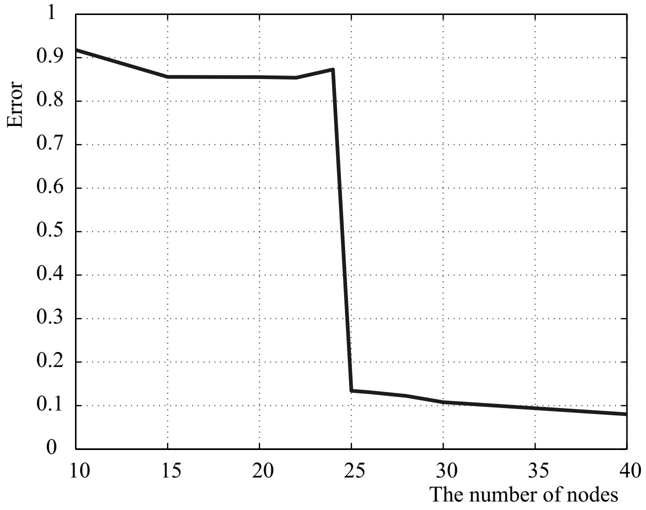

Fig. 18. Error value E_θ and the number of nodes.

an object by eye, throwing a ball, and so on. It is desirable that some abstracted behavior units are designed, and HMMs are applied to such behavior units. For this issue, we plan to construct hierarchical HMMs to be applied from simple motion level to complex behavior level.

We think this result is the first step to connect language development process to the motion acquisition process using the mimesis model; for instance, humanoids try to make communications with others, and build a relationship representation between proto-symbols and linguistic symbols. For such a direction, we try to define the distance between each HMM and to establish a computational method in order for the proto-symbols to evolve into general symbols. We believe that this approach leads the building of an intelligent system which connects humanoid intelligence and behavior science.

Appendix

A.1. Viterbi Algorithm for Motion Recognition

$P(\boldsymbol{O}|\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$ is calculated using the following equation which is called the "Viterbi algorithm". Let the forward probability $\alpha_j(t)$ for some model $\mathcal{P}_{\mathcal{S}}$ be defined as

$$\alpha_j(t) = P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, x(t) = j|\mathcal{P}_{\mathcal{S}}). \qquad (18)$$

That is, $\alpha_j(t)$ is the joint probability of observing the first t motion elements and being in state j at time t. This forward probability can be efficiently calculated by the following recursion:

$$\alpha_1(i) = 1 \tag{19}$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{o}_{t+1}) \tag{20}$$

$$\alpha_j(i) = a_1 b_j(\boldsymbol{o}_1) \tag{21}$$

$$P(\boldsymbol{O}|\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{N} \alpha_i(T). \tag{22}$$

A.2. Learning of Discrete Hidden Markov Model Parameters

To calculate the HMM parameters $\boldsymbol{A} = \{a_{ij}\}$, $\boldsymbol{B} = \{b_{ij}\}$ when the observation sequence $\boldsymbol{O}$ is given,

$$\gamma_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)} \tag{23}$$

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i, j) \tag{24}$$

are first defined. After this, new parameters are estimated using the following EM algorithms:

$$\hat{\pi}_i = \gamma_1(i) \tag{25}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{26}$$

$$\hat{b}_{i(k)} = \frac{\sum_{t:o_t=k} \gamma_t(i)}{\sum_{t=1}^{\mathrm{T}} \gamma_t(i)}. \tag{27}$$

After this, a parameter update is executed using the following equations. The inferences by eqs. (25), (26), and (27) are repeated until the value is converged:

$$\pi = \hat{\pi} \tag{28}$$

$$a_{ij} = \hat{a}_{ij} \tag{29}$$

$$b_{i(k)} = \hat{b}_{i(k)}. \tag{30}$$

The above processes are called Baum–Welch algorithms.

A.3. Learning of Continuous Hidden Markov Models

In the case of CHMMs, Baum–Welch algorithms are used as well as DHMMs.

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{\mathrm{T}} L_{jm}(t) \boldsymbol{o}_t}{\sum_{r=1}^{R} \sum_{t=1}^{\mathrm{T}} L_{jm}(t)} \tag{31}$$

$$\hat{\boldsymbol{\Sigma}}_{jm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{\mathrm{T}} L_{jm}(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_j)(\boldsymbol{o}_t - \boldsymbol{\mu}_j)'}{\sum_{r=1}^{R} \sum_{t=1}^{\mathrm{T}} L_{jm}(t)} \tag{32}$$

$$\hat{\boldsymbol{c}}_{jm} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{\mathrm{T}} L_{jm}(t)}{\sum_{r=1}^{R} \sum_{t=1}^{\mathrm{T}} L_{jm}(t)} \tag{33}$$

where

$$L_j(t) = \frac{1}{P(\boldsymbol{O}|\boldsymbol{A}, \boldsymbol{B})} \alpha_j(t) \beta_j(t) \tag{34}$$

$$\alpha_j(t) = \left\{\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij}\right\} b_j(\boldsymbol{o}_t) \tag{35}$$

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_j(t+1) \tag{36}$$

with the initial condition

$$\alpha(1) = 1 \tag{37}$$

$$\beta(1) = \sum_{j=2}^{N-1} a_{1j} b_j(\boldsymbol{o}_1) \beta_j(1). \tag{38}$$

Acknowledgments

This research was supported by the “Robot Brain Project” under the Core Research for Evolutional Science and Technology (CREST program) of the Japan Science and Technology Corporation. The authors would like to thank Mr K. Kurihara for assisting of the motion capturing system.

References

Deacon, T. W. 1997. *The Symbolic Species*, W.W. Norton & Co., New York.

Donald, M. 1991. *Origins of the Modern Mind*, Harvard University Press, Cambridge, MA.

Gallese, V., and Goldman, A. 1998. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences* 2(12):493–501.

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. 1998. The development of Honda humanoid robot. *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 1321–1326.

Imai, S., Tokuda, K., and Kobayashi, T. 1995. Speech parameter generation from HMM using dynamic features. *Proceedings of ICASSP95*, Detroit, MI, pp. 660–663.

Inamura, T., Nakamura, Y., and Simozaki, M. 2002. Associative computational model of mirror neurons that connects missing link between behaviors and symbols. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland, pp. 1032–1037.

Kobayashi, T., Masuko, T., Tokuda, K., and Imai, S. 1996. Speech synthesis from HMMs using dynamic features. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, pp. 389–392.

Kuniyoshi, Y., Inaba, M., and Inoue, H. 1994. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation* 10(6):799–822.

Kurihara, K., Hoshino, S., Yamane, K., and Nakamura, Y. 2002. Optical motion capture system with pan-tilt camera tracking and real-time data processing. *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, pp. 1241–1248.

Kuroki, Y., Ishida, T., Yamaguchi, J., Fujita, M., and Doi, T. 2001. A small biped entertainment robot. *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, Tokyo, Japan, pp. 181–186.

Matarić, M. J. 2000. Getting humanoids to move and imitate. *IEEE Intelligent Systems* 15(4):18–24.

Miyamoto, H., and Kawato, M. 1998. A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks* 11:1331–1344.

Morita, M. 1996. Memory and learning of sequential patterns by non-monotone neural networks. *Neural Networks* 9(8):1477–1489.

Morita, M., and Murakami, S. 1997. Recognition of spatiotemporal patterns by non-monotone neural networks. *Proceedings of the 1997 International Conference on Neural Information Processing*, Dunedin, New Zealand, Vol. 1, pp. 6–9.

Nishiwaki, K., Sugihara, T., Kagami, S., Kanehiro, F., Inaba, M., and Inoue, H. 2000. Design and development of research platform for perception-action integration in humanoid robot: H6. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Takamatsu, Japan, Vol. 3, pp. 1559–1564.

Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K. 2002. Modeling manipulation interactions by hidden Markov models. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland, pp. 1096–1101.

Okada, M., Tatani, K., and Nakamura, Y. 2002. Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion. *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, pp. 1410–1415.

Pook, P. K., and Ballard, D. H. 1993. Recognizing teleoperated manipulations. *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA, pp. 578–585.

Samejima, K., Katagiri, K., Doya, K., and Kawato, M. 2002. Symbolization and imitation learning of motion sequence using competitive modules. *Transactions of the Institute of Electronics, Information and Communication Engineers* J85-D-II(1):90–100.

Samejima, K., Doya, K., and Kawato, M. 2003. Intra-module credit assignment in multiple model-based reinforcement learning. *Neural Networks* 16:985–994.

Schaal, S. 1999. Is imitation learning the way to humanoid robots? *Trends in Cognitive Sciences* 3(6):233–242.

Tani, J. 2002. On the dynamics of robot exploration learning. *Cognitive Systems Research* 3(3):459–470.

Wada, T., and Matsuyama, T. 1998. Appearance based behavior recognition by event driven selective attention. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 759–764.

Yamato, J., Ohya, J., and Ishii, K. 1992. Recognizing human action in time-sequential images using hidden Markov model. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, IL, pp. 379–385.

Yoshiike, T., Konno, A., Nagashima, K., Inaba, M., and Inoue, H. 1998. On-line recognition and mimicking of human posture. *Proceedings of the 3rd International Conference on Advanced Mechatronics*, Japan, pp. 430–435.

Young, S. et al. 2000. *The HTK Book*, Microsoft Corporation.