2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Mimesis Scheme using a Monocular Vision System on a Humanoid Robot

Dongheui Lee and Yoshihiko Nakamura
Department of Mechano-Informatics
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, JAPAN
{*dhlee, nakamura*}*@ynl.t.u-tokyo.ac.jp*

***Abstract*— Optical motion capturing systems are widely used to acquire human beings' motion patterns in humanoid imitation learning research. However, optical motion capturing systems have a restricted movable area. This paper proposes the HMM based mimesis scheme using a monocular camera mounted on a humanoid. This scheme releases the restriction of movable area and enables imitation in daily life environments. Also, natural human-robot-interaction is expected during imitation. From two-dimensional image sequences of the demonstrator's motion, the demonstrator's pose and motion is estimated and recognized through the mimesis model and the humanoid generates its joint motor commands for imitation in 3D space. The feasibility of the proposed scheme is demonstrated by simulation.**

***Index Terms*— monocular vision, mimesis model, proto-symbol, partial observations, Multidimensional scaling**

I. INTRODUCTION

Imitation skills for humanoids have received a great deal of attention because the imitation function is the most primitive and fundamental factor of intelligence. Neuroscience based evidence of motor primitives and mirror neurons [1] inspired the development of corresponding forms of robot imitation learning. Bentivegna and Atkeson [2] used the idea of primitives for motor learning. Billard and Mataric [3] used connectionist-based approaches to represent movements. Inamura et al. [4] proposed the mimesis model which integrates a bidirectional framework of motion recognition and motion generation through embedded symbols of whole body dynamics.

Motion capturing systems are widely used [4] [5] [6] [7] to acquire reference motion patterns, such as human beings' motion patterns in research on humanoid imitation learning. Most motion capturing systems use optical devices, consisting of passive optical markers and multiple cameras [8]. However, they are inconvenient in daily life environments because of the restrictions on the movable area. Some imitation research [7] adopts wearable motion capturing systems. The wearable types solve the problem of movable area restriction. However, since human robot interaction is greatly important in imitation, in order to achieve natural human robot interaction, a mimesis model using a simple vision system on the humanoid is needed.

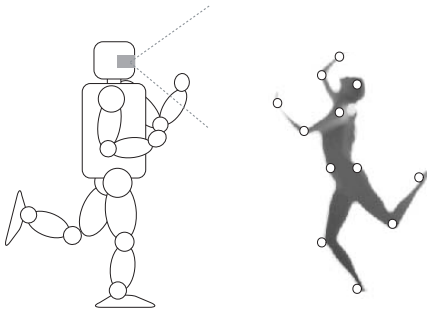

Fig. 1. mimesis with a monocular vision system

This paper proposes the HMM based mimesis model which uses only a camera on a humanoid, instead of optical or wearable motion capture systems (Fig. 1). From 2D image sequences (especially labeled markers on the demonstrator body) of the demonstrator's motion, the humanoid recognizes the observed motion and imitates the motion in 3D space by generating joint motor commands through embedded symbols.

With this strategy, the following advantages are achieved. (1) Imitation learning becomes possible in daily life environments without restrictions on the movable area. (2) Since the monocular or binocular vision systems are built into most humanoid robots, the proposed method does not require extra systems like an optical motion capturing system [8] or a wearable motion capturing system [7]. (3) By using the humanoid onboard vision system, human robot interaction becomes more natural during imitation.

Comparing with the conventional optical motion capturing systems, onboard monocular or binocular vision systems do not provide enough depth information. Among the research dealing with incomplete data, Ghahramani and Jordan [9] proposed using the expectation-maximization (EM) algorithm to fill in missing feature values of examples when learning from incomplete data by assuming a mixture model. The authors proposed a imitation method of whole body motion from partial observations in [5]. However, these methods [5] [9] are limited to dealing with incomplete data in the same space. Since general mimesis from partial observations is a

1-4244-0602-1/07/$20.00 ©2007 IEEE.

nonlinear problem, this paper proposes a nonlinear mimesis solution, especially for the case that the partial observations are perspective projected observations of a single camera. In the proposed scheme, the positions and postures of a demonstrator are not known a-priori but are estimated by adopting the multidimensional scaling method [10].

Among a number of works on human pose estimation [11] [12][13] using vision systems, Agarwal and Triggs [13] propose a method of 3D human pose recovery from 2D images, using silhouettes, but the experimental results are limited to pose estimation during walking. Comparing with this work [13], our method covers various human motions and recognizes motions as well as human poses.

Before proceeding to the algorithm description, coordinates' terms are introduced. ${}^{base}o_{space}$ denotes that the observation o is represented in the coordinates whose base is $base$ and whose description type is $space$. There are three types of $base$s: $\{C\}$, $\{I\}$ and $\{D\}$ indicate "Camera origin", "camera's projected Image" and "Demonstrator's base body". For the $space$, θ, x, and ϕ denote "joint angle space", "cartesian space", and "spherical space." The coordinates are named as "$base\ space$ coordinates". For example, ${}^{D}o_{\theta}$, ${}^{C}o_{\phi}$, and ${}^{I}o_{x}$ denote observation in the "demonstrator joint coordinates", "camera spherical coordinates", and "image cartesian coordinates" respectively.

II. Mimesis from Perspective Projected Observations of a Monocular Camera

A. Problem Statement

The target problem is the nonlinear form of complete mimesis from incomplete observations: imitating similar motion patterns in the 3D space from observing 2D camera images of target motion patterns. In other words, after watching 2D pixel information of labeled markers attached to a human, a humanoid produces the motion.

Mimesis Model: The HMM based mimesis model [4] is adopted. HMM based mimesis model consists of three functions: learning, recognition and generation. Hidden Markov Model (HMM) is used as the mathematical backbone for such integration. Learning means the emergence of a proto-symbol which represents the dynamics of motion sequences. Recognition finds the most likely proto-symbol for observation. The generation function decodes motion patterns from the proto-symbol.

Proto-symbols: A proto-symbol is defined as the parameters of a Hidden Markov Model $\lambda = \{A, B, \pi\}$, where A is the state transition probability matrix and π is the initial state probabilities vector. The observation symbol probability distribution B is represented with a mixtur of Gaussian distributions $B = \{c, \mu, \Sigma\}$, where c is the weight of the mixture component, μ is the mean vector, and Σ is the covariance matrix. Proto-symbols are learned from motion patterns, which are represented in the joint angle space $\lambda_\theta = \{A_\theta, \pi_\theta, c_\theta, \mu_\theta, \Sigma_\theta\}$, by the EM algorithm. Thus, the size of the vector $\mu_\theta \in R^M$ and matrix $\Sigma_\theta \in R^{M\times M}$ is the number of joint angles.

Observations: The observed motion patterns ${}^{I}o_x$ are represented as pixel positions of labeled markers, which are attached on fixed positions of a demonstrator body, on the camera image. The observation is the 2D Cartesian value of the visible markers from a monocular camera ${}^{I}o_x \in R^{2N}$, where N is the number of visible markers. For generality, the observations are based on perspective projection and all the markers are not always visible. However, it is assumed that the labels of markers are known.

To summarize, the dimensions of the vector ${}^{I}o_x$ and that of vector μ_θ and matrix Σ_θ are different. ${}^{I}o_x$ and λ_θ are based in the image cartesian coordinates and in the joint coordinates respectively. Also, the position and posture of the humanoid and the demonstrator are not given. It is assumed that internal camera parameters are known but external parameters are unknown. Under such conditions, in order to solve the mimesis of complete motion patterns from image sequences of a single camera, the following two strategies are proposed.

B. Strategy I

(I) Proto-symbols are converted as ${}^{D}\lambda_\theta \rightarrow {}^{D}\lambda_x \rightarrow {}^{C}\lambda_x \rightarrow {}^{I}\lambda_x$. Proto-symbols in the demonstrator joint coordinates are converted into the demonstrator cartesian coordinates, into the camera cartesian coordinates, and into the image cartesian coordinates.

${}^{D}\lambda_\theta \rightarrow {}^{D}\lambda_x$: The conversion from "demonstrator joint coordinates" into the "demonstrator cartesian coordinates" by kinematics is given in section III-A. Proto-symbols are converted by the nonlinear kinematics function via Monte Carlo method (section IV).

${}^{D}\lambda_x \rightarrow {}^{C}\lambda_x$: The transformation matrix between the camera coordinates and the demonstrator coordinates is found by applying the multidimensional scaling algorithm (section V-A). Then, ${}^{D}\lambda_x$ is transformed into the camera coordinates ${}^{C}\lambda_x$ by the transformation matrix. (section V-B)

${}^{C}\lambda_x \rightarrow {}^{I}\lambda_x$: Perspective projection of a marker in three-dimensional space into the camera image is summarized in the section III-B. Because this perspective projection considering camera distortion is a nonlinear conversion, conversion of proto-symbols are carried out in a similar way to section IV.

(II) Recognition and generation: Finally, both proto-symbols ${}^{I}\lambda_x$ and observations ${}^{I}o_x$ are represented in the 2D image Cartesian coordinates. When all the markers are not visible, linear mimesis problem from partial observations are carried out as described in section VI. The details can be found in [5].

C. Strategy II

(I) Proto-symbols are converted as ${}^{D}\lambda_\theta \rightarrow {}^{D}\lambda_x \rightarrow {}^{C}\lambda_x \rightarrow {}^{C}\lambda_\phi$.

${}^{C}\lambda_x \rightarrow {}^{C}\lambda_\phi$: Conversion of a vector from cartesian coordinates to spherical coordinates is summarized in the section III-D. Since this conversion is nonlinear, proto-symbols are converted in the same way as in section IV.

(II) Observation is converted as ${}^{I}o_x \rightarrow {}^{C}o_x \rightarrow {}^{C}o_\phi$.

${}^{I}o_x \rightarrow {}^{C}o_x$: From a projected marker position, the real position in three-dimension is estimated by the section III-C.

(III) Recognition and generation: Then, both proto-symbols ${}^{C}\lambda_\phi$ and observations ${}^{C}o_\phi$ are represented in the camera spherical coordinates. Because a monocular camera does not provide depth information, this becomes the linear mimesis problem of complete motion pattern from incomplete observations. Recognition and generation of motions is carried out by the method in the section VI.

III. Basic Conversion Types

A. Forward Kinematics

The kinematic model of the humanoid is given in Fig. 2 and eq. (1). In Fig. 2, the left figure shows 20 joint angles o_θ and the right figure shows the 16 markers' cartesian position (x, y, z) o_x.

$$o_x = f(o_\theta) \tag{1}$$

where $o_x \in R^{48}$, $o_\theta \in R^{20}$.

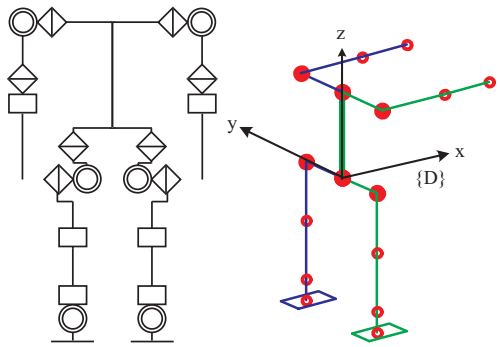

Fig. 2. Humanoid configuration (left: Joint angle ($o_\theta \in R^{20}$), right: Markers' Cartesian coordinates ($o_x \in R^{48}$))

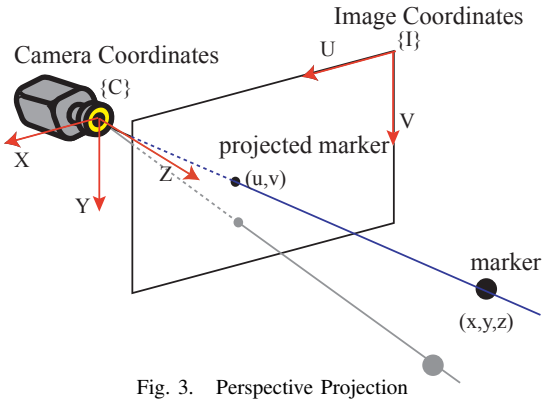

Fig. 3. Perspective Projection

B. Perspective Projection - from 3D to 2D

Based on perspective projection (Fig. 3), a positional vector (x, y, z) in 3-dimensional camera coordinates is projected into a 2D pixel positional vector (u, v) on the projected image coordinates as follows, by considering image distortion.

A scaling factor k is calculated as $k = \frac{z}{\alpha}$, where α is the camera focal length, the distance between camera origin and the projection plane.

$$(dx, dy, dz) = \frac{1}{k}(x, y, z) = (\frac{x}{k}, \frac{y}{k}, \alpha) \tag{2}$$

$$u = \frac{2dx}{s_u(1 + \sqrt{1 - 4\kappa(dx^2 + dy^2)})} + c_u \tag{3}$$

$$v = \frac{2dy}{s_v(1 + \sqrt{1 - 4\kappa(dx^2 + dy^2)})} + c_v \tag{4}$$

where, (c_u, c_v) is the center pixel position on the image and s_u and s_v are the size of a pixel in the u and v direction. κ is the camera distortion parameter.

C. Perspective Projection - from 2D to 3D

In the opposite direction, a 2D pixel positional vector (u, v) on the projected image coordinates can be converted into a 3D vector (x, y, z) in camera cartesian coordinates.

$$(\tilde{u}, \tilde{v}, \alpha) = (s_u(u - c_u), s_v(v - c_v), \alpha) \tag{5}$$

$$(dx, dy, \alpha) = (\frac{\tilde{u}}{1 + \kappa(\tilde{u}^2 + \tilde{v}^2)}, \frac{\tilde{v}}{1 + \kappa(\tilde{u}^2 + \tilde{v}^2)}, \alpha) \tag{6}$$

$$(x, y, z) = k(dx, dy, \alpha) \tag{7}$$

where the scaling factor k is arbitrary.

D. Spherical Coordinate

Spherical coordinates are also called 3D polar coordinates. A vector (x, y, z) in cartesian coordinates becomes (ρ, θ, ϕ) into spherical coordinates where the radius is $0 \leq \rho$, colatitude is $0 \leq \phi \leq \pi$, and longitude is $0 \leq \theta \leq 2\pi$.

$$(\rho, \theta, \phi) = (\sqrt{x^2 + y^2 + z^2}, tan^{-1}(\frac{y}{x}), tan^{-1}(\frac{\sqrt{x^2 + y^2}}{z})) \tag{8}$$

IV. Proto-Symbol Conversion by Kinematics

This section considers how to convert a proto-symbol λ from the joint space $\lambda_\theta = \{A_\theta, \pi_\theta, B_\theta\}$ to the Cartesian space $\lambda_x = \{A_x, \pi_x, B_x\}$ by forward kinematics (section III-A). Here it is assumed that after converting mixture gaussian distributions $B_\theta = \{c_\theta, \mu_\theta, \Sigma_\theta\}$ to the Cartesian space, the converted motion output probability distributions $B_x = \{c_x, \mu_x, \Sigma_x\}$ are still a gaussian distribution.

Since the main difference between λ_θ and λ_x is the representation of motion patterns, A, π, and c parameters, which are not directly related to the observations, do not need to be changed. $A_x = A_\theta$. $\pi_x = \pi_\theta$. $c_x = c_\theta$.

A. Conversion of μ and Σ, When Covariance is Small

When the forward kinematic model is given by eq. (1) and the covariance is small, the mean vector μ_x is calculated from μ_θ by eq. (9). The covariance matrix is converted by eq. (10), by using the Jacobian matrix of the forward kinematics. In particular, the Jacobian matrix of the mean vector is applied.

$$\mu_x = f(\mu_\theta) \quad (9)$$

$$\Sigma_x = J(\mu_\theta)\Sigma_\theta J(\mu_\theta)^T \quad (10)$$

where

$$J(\theta) = \frac{\partial f(\theta)}{\partial \theta} \quad (11)$$

Although the kinematic model is a nonlinear function, if the covariance is small enough, it can be approximated as a linear function. Our experimental data shows that most of the covariance matrix elements $\Sigma_{\theta ij}$ of proto-symbols λ_θ are 0.015. Therefore, the standard deviation is 0.122 rad (7 deg). This can be roughly considered as a linear function.

B. Conversion of μ and Σ, When Covariance is Large

When the covariance is large, the mean vector μ_x and covariance matrix Σ_x are calculated by Monte Carlo method. The Monte Carlo method estimates the continuous probability distribution function by using discrete particles. Particles $o_{\theta i}$ of o_θ are generated from the $B_\theta = \{c_\theta, \mu_\theta, \Sigma_\theta\}$. $o_{\theta i} \in R^{20}$ denotes the i-th particle of o_θ, where $i = 1, ..., nS$ and nS is the number of particles. Each particle is converted from the joint space to the Cartesian space by kinematics.

$$o_{xi} = f(o_{\theta i}) \quad (12)$$

$o_{xi} \in R^{48}$ denotes the i-th particle of o_x. Then, the converted mean vector and covariance matrix are calculated.

$$\mu_x = \frac{1}{nS}\sum_i o_{xi} \quad (13)$$

$$\Sigma_x = \frac{1}{nS}\sum_i (o_{xi} - \mu_x)(o_{xi} - \mu_x)^T \quad (14)$$

Calculation via Monte Carlo method is simpler because the Jacobian matrix is not necessary. Its computational cost is proportional to the desired accuracy.

V. Proto-Symbol Conversion by Transformation Matrix

A. Transformation Matrix Search

Position and rotation of camera coordinates $\{C\}$ and demonstrator coordinates $\{D\}$ are not specified a-priori. In the given kinematic model (Fig. 2), six markers on the waist, chest, left shoulder, right shoulder, left hip and right hip are static with respect to the demonstrator's coordinates $\{D\}$. The six markers are displayed with bold red circles in Fig. 2. By using three markers' positions, the transformation matrix ${}^D_C T$ can be calculated. Here, three position vectors p_0 (waist), p_1 (left hip), p_2 (chest) are used, as shown in Fig. 4.

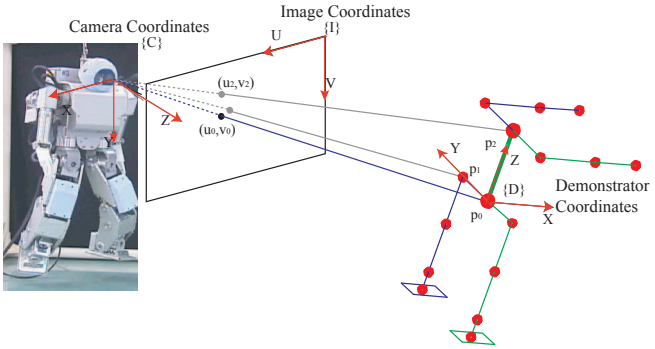

Fig. 4. Transformation Matrix Search

The i-th marker position $p_i = (x_i, y_i, z_i)$ for $i = 0, 1, 2$ is estimated from its projected 2-dimensional pixel position (u_i, v_i) on the image. The projected point (u_i, v_i) on the image is represented as (dx_i, dy_i, α) with respect to camera coordinates (eq. (6)), where α is the camera focal distance. A real marker position $p_i = (x_i, y_i, z_i)$ is on the line which connects the camera origin and the projected point (dx_i, dy_i, α) on the projection plane.

$$p_i = (x_i, y_i, z_i) = k_i(dx_i, dy_i, \alpha) \quad (15)$$

First, a rough scaling factor k is defined as

$$k = min\frac{D(p_{im}, p_{jm})}{D(p_i, p_j)} \quad (16)$$

, where $D(p_{im}, p_{jm})$ is the Euclidean distance between marker i and j in the model and $D(p_i, p_j)$ is the calculated Euclidean distance between marker i and j. Every scaling factor k_i is set as the rough scaling factor. $k_i = k$. p_i is updated as $p_i = k_i(dx_i, dy_i, \alpha)$.

Second, a precise scaling factor of i-th marker k_i is updated by the multidimensional scaling method. The evaluation function is defined.

$$W(h) = \sum_j (D(hp_i, p_j)^2 - D(p_{im}, p_{jm})^2)^2 \quad (17)$$

The value of h which minimizes the evaluation function $W(h)$ is found via the successive over relaxation (SOR) method. The scaling factor k_i is updated by the following equation.

$$k_i \leftarrow hk_i \quad (18)$$

The real 3D marker position is estimated as $p_i = k_i(dx_i, dy_i, \alpha)$. For other relative static markers, their optimal scaling factors are founded and 3D marker positions are estimated in the same way.

Last, the transformation matrix is estimated: The translation is the waist position p_0. The unit vectors of $p_1 - p_0$ and $p_2 - p_0$ are corresponding to the Y axis and Z axis of $\{D\}$ coordinates. By cross product, the unit vector of X axis is

known.

$$ {}^{C}p_{Dorg} = p_0 \tag{19} $$

$$ e_y = \frac{p_1 - p_0}{|p_1 - p_0|},\ e_z = \frac{p_2 - p_0}{|p_2 - p_0|},\ e_x = e_y \times e_z \tag{20} $$

$$ {}^{C}_{D}R = [\ e_x \quad e_y \quad e_z\ ] \tag{21} $$

$$ {}^{C}_{D}T = \begin{bmatrix} {}^{C}_{D}R & {}^{C}p_{Dorg} \\ 0 & 1 \end{bmatrix} \tag{22} $$

This algorithm works even when all the markers are not visible, such as when the demonstrator's back is totally not visible and his front is visible in the 2D space. Because the transformation matrix is founded with only three markers, the minimum working condition is that three static markers in $\{D\}$ coordinates are partially visible.

Also, as long as the ratio of body structure is same, the method can be applied to other subjects with a variety of height. This is because the scaling factors k_i are adjusted with respect to the body size and the distance between the camera and demonstrator.

B. Conversion Symbols with Transformation Matrix

With the obtained transformation matrix, proto-symbols ${}^{D}\lambda_x = \{{}^{D}A_x, {}^{D}\pi_x, {}^{D}B_x\}$, which is estimated in section IV, in the demonstrator cartesian coordinates is converted into those ${}^{C}\lambda_x = \{{}^{C}A_x, {}^{C}\pi_x, {}^{C}B_x\}$ in the camera cartesian coordinates. As with the conversion of λ_θ to λ_x in section IV, A, π, and c parameters are not changed. ${}^{C}A_x = {}^{D}A_x$. ${}^{C}\pi_x = {}^{D}\pi_x$. ${}^{C}c_x = {}^{D}c_x$.

Because coordinates transformation is a general linear system, mean vector and covariance matrix are calculated as follows.

$$ \begin{bmatrix} {}^{C}o_{xi} \\ 1 \end{bmatrix} = {}^{C}_{D}T \begin{bmatrix} {}^{D}o_{xi} \\ 1 \end{bmatrix} \tag{23} $$

$$ \begin{bmatrix} {}^{C}\mu_{xi} \\ 1 \end{bmatrix} = {}^{C}_{D}T \begin{bmatrix} {}^{D}\mu_{xi} \\ 1 \end{bmatrix} \tag{24} $$

$$ {}^{C}\Sigma_{xi} = {}^{C}_{D}R\,{}^{D}\Sigma_{xi}\,{}^{C}_{D}R^{T} \tag{25} $$

where $o_{xi} \in R^3$, $\mu_{xi} \in R^3$ and $\Sigma_{xi} \in R^{3\times3}$ $(i = 1, \cdots, 16)$ are the position vector, mean vector and covariance matrix of the i-th marker's 3D Cartesian coordinates. Because the covariance matrix is not related to the translation vector, the covariance matrix is calculated by considering the rotation matrix ${}^{C}_{D}R$.

VI. Linear Mimesis Problem from Partial Observations

A. Motion Recognition

The observed motion is recognized by searching the most probable HMM for input observation sequences among HMMs, by calculating the likelihood $P(x|\lambda)$.

$$ \lambda^* = \arg\max_{\lambda} P(x|\lambda) \tag{26} $$

This section explains how to calculate the likelihood $P(x|\lambda)$ in the case that there are missing motion elements $\{x_k\}_t$ in input observation sequences x_t. The likelihood that a proto-symbol λ generates the observed motion x is computed by the forward algorithm [14].

$$ P(x|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{27} $$

where $\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(x_{t+1})$, whose initialization is $\alpha_1(i) = \pi_i b_i(x_1)$. $\alpha_t(i)$ is the forward variable, which denotes the probability of the observation $o_1, o_2, \cdots o_t$ and i-th state at time t. $b_i(x_t)$ is the probability density function for the output of continuous vector x_t at the i-th state node. It is represented with a mixture of Gaussian distributions.

$$ b_i(x_t) = \sum_{j=1}^{m} c_{ij} b_{ij}(x_t) \tag{28} $$

$$ b_{ij}(x_t) = \frac{exp\{-\frac{1}{2}(x_t - \mu_{ij})^T \Sigma_{ij}^{-1}(x_t - \mu_{ij})\}}{\sqrt{(2\pi)^M \det \Sigma_{ij}}} \tag{29} $$

For the missing motion elements, the following equation is applied into eq. (29), $\{x_k\}_t - \mu_{ij} = const$, so that the invisible motion elements do not affect the output probability density function with any proto-symbols. In our simulations, the constant value is set to zero.

B. Motion Generation

Motion patterns are decoded using the expectation operator in the stochastic model. The motion generation is a two-stage stochastic process: state transition and motion output.

(I) classical motion generation method: When generating a motion pattern only based on the proto-symbol, eq. (30) is used. State transition is generated by A and π. Motion output is generated by B.

$$ y = g(\lambda^*) \tag{30} $$

(II) observation conditioned motion generation method: When generating a similar motion pattern to the observed motion pattern, eq. (31) is used.

$$ y = g(\lambda^*, x) \tag{31} $$

The state sequence is obtained by applying the Viterbi algorithm [14], which finds the single best state sequence for the given observation sequence. Thus, this optimal state transition generation enables us to generate a motion pattern close to the observed target motion pattern. Here also, for the invisible motion elements, $x - \mu = const$ is applied. After the optimal state sequence is obtained, the output observation sequence y is calculated according to the output probability distribution in state i, i.e., $b_i(x)$.

VII. SIMULATIONS

This section shows the simulation results of the proposed approach. The humanoid possesses following nine proto-symbols: (1) walk, (2) raising two arms (raise arms), (3) dance, (4) kick, (5) punch, (6) sumo stomp, (7) squat, (8) throw, and (9) bending upper body forward (bend forward). When learning the proto-symbols by the EM algorithm, joint angular data (20 DOF) are used. For the continuous Hidden Markov Model, the number of nodes is 20 and that of mixture Gaussians is 3. For each proto-symbol, 13∼28 motion patterns are used as a training set where each motion pattern is about 2 second motion.

Input data for mimesis is the image sequences of 16 markers' pixel positions on the image from an arbitrary view. This is corresponding to perspective projected observation of a single camera. Above nine motions are observed from three different views. Each motion is recognized by strategy I and strategy II. Proto-symbols are converted into an appropriate coordinates. If necessary, the observation sequences are also converted into the same coordinates. Finally, the linear mimesis problem from partial observation is carried out. The most likely proto-symbol is found. The humanoid generates similar motion to the target motion in the three-dimensional space.

In Fig. 5 and Fig. 6 show some mimesis results by strategy I. Fig. 5 shows results of walk, raising two arms, and dance, when the demonstrator's pose (position and euler angles) is $(x, y, z, \alpha, \beta, \gamma) = (0, 0.3, 1.0, -45, 0, 0)$ with respect to camera coordinate. The units of position (x, y, z) and euler angles (α, β, γ) are m and $degree$. Fig. 6 shows mimesis results of kick, sumo stomp, and squat, when the demonstrator's pose is $(x, y, z, \alpha, \beta, \gamma) = (0, 0.3, 1.0, 90, -45, 0)$ with respect to camera coordinate. Mimesis results by strategy II are shown in Fig. 7. "throw" and "bend upper body forward" motion image sequences are observed like (a) and (c), when the demonstrator's pose is $(x, y, z, \alpha, \beta, \gamma) = (-0.2, 0.1, 1.1, 90, 53, 0)$ with respect to camera coordinate. Corresponding generated motions are (b) and (d).

From simulation results, the averaged demonstrator pose estimation error, mimesis error and motion recognition success percentages are calculated and shown in Table I. Position error is the error of demonstrator's 3D position estimation error. Yaw angle error is the demonstrator's yaw value estimation error. Joint angle error shows the averaged error per joint, which is the difference between the observed and generated joint angles. When an observed motion is recognized as a correct proto-symbol, it is counted in for motion recognition success.

VIII. CONCLUSION AND FUTURE WORK

In this paper, probabilistic mimesis from partial observation is extended into a nonlinear problem; motion imitation in the 3D space from image sequences of a monocular camera. In order to solve this problem, important strategies are as follows. (1) By applying conversion rules, proto-symbol and observation are converted into the same coordinates. Typical linear conversion of proto-symbols is addressed in section V. Also, a case of nonlinear conversion is shown in section IV. (2) The transformation matrix between the camera's coordinates and the demonstrator's coordinates is estimated by the multidimensional scaling method. (3) Motion recognition and generation is carried out by the linear mimesis method from partial observations.

This paper shows that a humanoid robot can understand and imitate human motion in daily life by using only the onboard vision system. However, some improvements are required. Currently, our method is based on labeled markers and this paper shows only simulation results. For the future

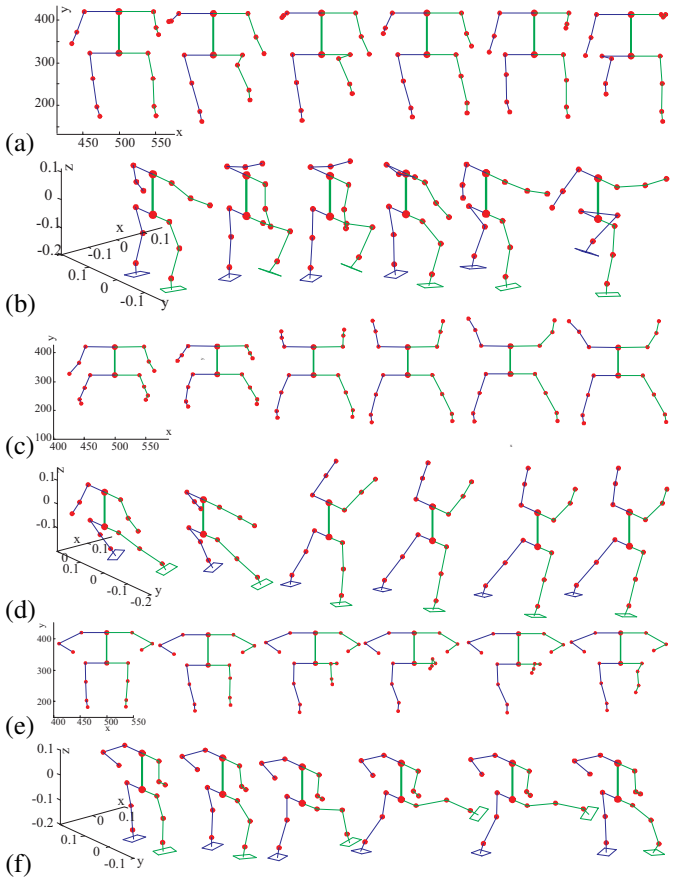

Fig. 5. Mimesis by Strategy I: (a) "walk" 2D observation, (b) "walk" 3D generated motion, (c) "raise arms" 2D observation, (d) "raise arms" 3D generated motion, (e) "dance" 2D observation, (f) "dance" 3D generated motion

TABLE I
POSE ESTIMATION ERROR AND IMITATION ERROR

	error
position error	0.71 (mm)
yaw angle error	8.87 (deg)
joint angle error	5.42 (deg)
motion recognition success	96%

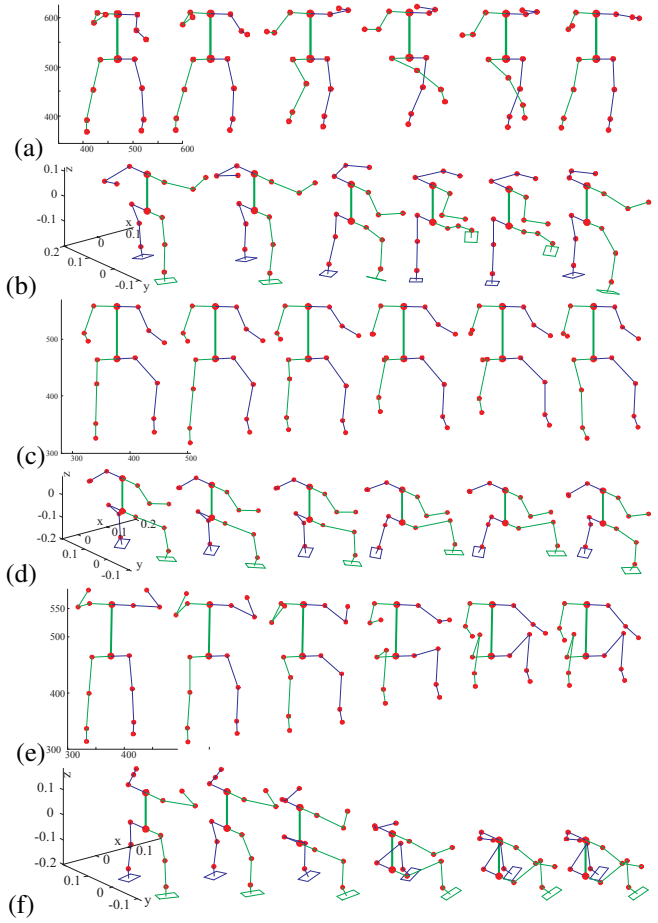

Fig. 6. Mimesis by Strategy I: (a) “kick” 2D observation, (b) ”kick” 3D generated motion, (c) “sumo stomp” 2D observation, (d) ”sumo stomp” 3D generated motion, (e) “squat” 2D observation, (f) ”squat” 3D generated motion

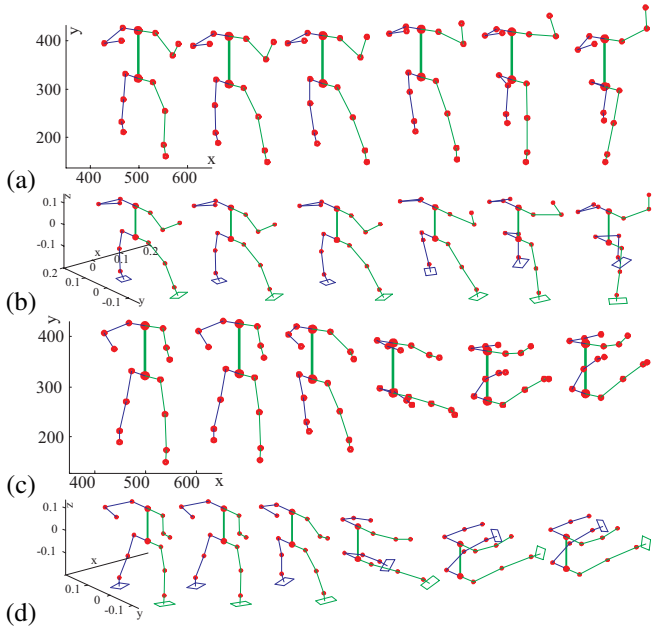

Fig. 7. Mimesis by Strategy II: (a) “throw” 2D observation, (b) “throw” 3D generated motion, (c) “bend forward” 2D observation, (d) ”bend forward” 3D generated motion

work, realtime mimesis experiments with a monocular camera will be carried out. Also, marker’s positional uncertainties which are caused by camera sensing errors and kinematics modeling errors should be considered in the probabilistic mimesis model. It is desirable not to use artificial markers.

ACKNOWLEDGMENT

This research was supported by Category S of Grant-in-Aid for Scientist Research, Japan Society for the Promotion of Science. The authors would like to thank professor Yamane for his advice and support on motion capture systems.

REFERENCES

[1] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi, “Premotor cortex and the recognition of motor actions,” *Cognitive Brain Research*, vol. 3, pp. 131–141, 1996.

[2] D. C. Bentivegna and C. G. Atkeson, “Using primitives in learning from observation,” in *First IEEE-RAS International Conference on Humanoid Robots (Humanoids 2000)*, 2000.

[3] A. Billard and M. J. Mataric, “Learning human arm movements by imitation: Evaluation of biologically inspired connectionist architecture,” *Robotics and Autonomous Systems*, vol. 37, pp. 145–160, 2001.

[4] T. Inamura, Y. Nakamura, and I. Toshima, “Embodied symbol emergence based on mimesis theory,” *International Journal of Robotics Research*, vol. 23, no. 4, 2004.

[5] D. Lee and Y. Nakamura, “Mimesis from partial observations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’05)*, Edmonton, Canada, August 2005, pp. 1911–1916.

[6] S. Nakaoka, A. Nakazawa, F. Kanahiro, K. Kaneko, M. Morisawa, and K. Ikeuchi, “Task model of lower body motion for a biped humanoid robot to imitate human dances,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’05)*, 2005, pp. 2769–2774.

[7] T. Inamura, N. Kojo, T. Sonoda, K. Sakamoto, K. Okada, and M. Inaba, “Intent imitation using wearable motion capturing system with on-line teaching of task attention,” in *IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, 2005, pp. 469–474.

[8] K. Kurihara, S. Hoshino, K. Yamane, and Y. Nakamura, “Optical motion capture system with pan-tilt camera tracking and realtime data processing,” in *IEEE International Conference on Robotics and Automation (ICRA’02)*, vol. 2, 2002, pp. 1241–1248.

[9] Z. Ghahramani and M. I. Jordan, “Supervised learning from incomplete data via an EM approach,” in *Advances in Neural Information Processing Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6. Morgan Kaufmann Publishers, Inc., 1994, pp. 120–127.

[10] M. Cox and M. Cox, *Multidimensional Scaling*. Chapman and Hall, 2001.

[11] J. K. Aggarwal and Q. Cai, “Human motion analysis: A review,” *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 3, pp. 428–440, 1999.

[12] L. Ren, G. Shakhnarovich, J. Hodgins, P. Viola, and H. Pfister, “Learning silhouette features for control of human motion,” 2004.

[13] A. Agarwal and B. Triggs, “Recovering 3d human pose from monocular images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, January 2006.

[14] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proc. IEEE*, pp. 257–286, 1989.