International Journal of Computer Vision
https://doi.org/10.1007/s11263-018-1123-1

Describing Upper-Body Motions Based on Labanotation for Learning-from-Observation Robots

Katsushi Ikeuchi[1] · **Zhaoyuan Ma**[2] · **Zengqiang Yan**[3] · **Shunsuke Kudoh**[4] · **Minako Nakamura**[5]

Received: 20 March 2017 / Accepted: 26 September 2018

Abstract

We have been developing a paradigm that we call learning-from-observation for a robot to automatically acquire a robot program to conduct a series of operations, or for a robot to understand what to do, through observing humans performing the same operations. Since a simple mimicking method to repeat exact joint angles or exact end-effector trajectories does not work well because of the kinematic and dynamic differences between a human and a robot, the proposed method employs intermediate symbolic representations, tasks, for conceptually representing what-to-do through observation. These tasks are subsequently mapped to appropriate robot operations depending on the robot hardware. In the present work, task models for upper-body operations of humanoid robots are presented, which are designed on the basis of Labanotation. Given a series of human operations, we first analyze the upper-body motions and extract certain fixed poses from key frames. These key poses are translated into tasks represented by Labanotation symbols. Then, a robot performs the operations corresponding to those task models. Because tasks based on Labanotation are independent of robot hardware, different robots can share the same observation module, and only different task-mapping modules specific to robot hardware are required. The system was implemented and demonstrated that three different robots can automatically mimic human upper-body operations with a satisfactory level of resemblance.

Keywords Humanoid robot · Programing-by-demonstration · Labanotation · Task model · Gesture recognition · Action recognition · Key pose

1 Introduction

In recent times, robotic applications have increased significantly. Traditionally, their applications have been limited to industrial domains. However, as of late, robots have been used in other areas, including family service (Wada and Shibata 2007), medical applications (Dombre et al. 2003; Roy et al. 2009) disaster management, and defense applications (Treptow et al. 2005; Aboshosha and Zell 2003). In response to this trend, a wider variety of robotic mechanisms have been developed, with increasing degrees of freedom (DOF). A representative example is the commercialization of humanoid robots, which typically have more than 40 degrees of freedom. For such high DOF, one of the imminent issues is programming operations in an efficient manner.

We have been working on the learning-from-observation paradigm to reduce the burden of programming (Ikeuchi et al. 1991; Ikeuchi and Suehiro 1994). If we can train a robot to perform a task by merely observing a human performing the same task, we can drastically decrease the programming cost.

Communicated by Tae-Kyun Kim, Stefanos Zafeiriou, Ben Glocker and Stefan Leutenegger.

✉ Katsushi Ikeuchi
katsuike@microsoft.com

Zhaoyuan Ma
zma3@wpi.edu

Zengqiang Yan
zyanad@connect.ust.hk

Shunsuke Kudoh
s-kudoh@uec.ac.jp

Minako Nakamura
nakamura.minako@ocha.ac.jp

[1] Microsoft Corporation, Redmond, USA

[2] Worcester Polytechnic Institute, Worcester, USA

[3] Hong Kong University of Science and Technology, Hong Kong, China

[4] University of Electro-Communications, Tokyo, Japan

[5] Ochanomizu University, Tokyo, Japan

Published online: 05 October 2018

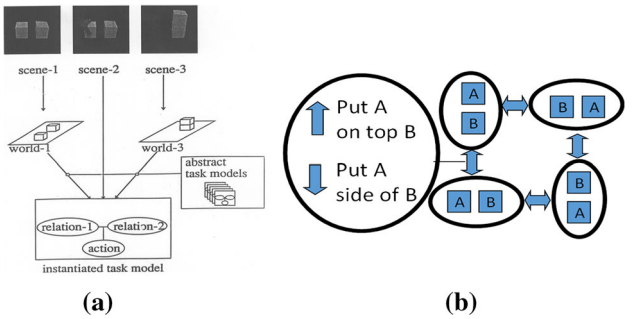

Fig. 1 Task recognition and state transitions. **a** Task recognition. Abstract task model associates one state transition with a necessary robot operation to create the transition and **b** state transitions and associated robot actions in the two-block world

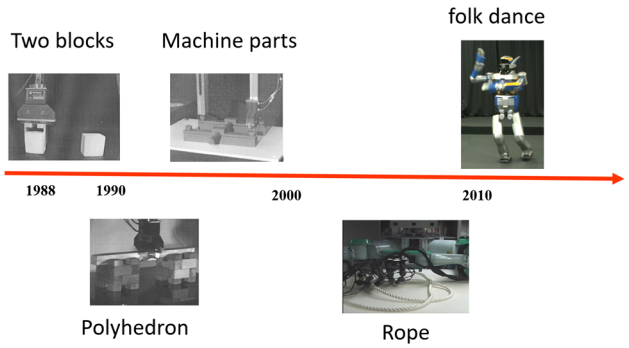

Fig. 2 Exploration of task domains

This paradigm is well suited for humanoid robots because of their high DOF making manually programming such operations very difficult if not impossible.

For the learning-from-observation paradigm, we previously proposed the task-and-skill framework. The framework separates common components (such as "what to do") referred to as tasks, from personal variegations ("how to do") referred to as skills (Ikeuchi et al. 1991; Ikeuchi and Suehiro 1994). A task is defined as a robot operation for achieving a goal by generating one specific state transition, and a skill is considered a trajectory among such state transitions and the speed variations along such trajectory (Ikeuchi and Suehiro 1994).

In this work, we focus on tasks. Each task is defined based on its goals, such as to achieve a specific contact state or to show a specific human posture, in the various domains of human activities. For an example, let us consider the assembly of a pair of cubes with the aim of achieving specific contact states, as shown in Fig. 1a. In this domain, two cubes, say A and B, are defined to have four states; "A on top of B", "B on top of A", "A to the left of B", and "B to the left of A" as shown in Fig. 1b. A task in this case, for example, is to create a transition of contact states between the two cubes. For instance, one transition is from the state "A to the left of B" to the state "A on top of B". To each state transition we can assign one necessary operation to result in the desired next state—in this case, the operation "Put-A-on-top-of-B". This association between a state transition and the necessary operation is defined as a task model (Ikeuchi et al. 1991).

We define our task recognition scheme as an extension of object recognition. In the offline mode of object recognition, abstract object models are prepared and stored in a computer's database. In the online mode, the computer associates model features with real features, identifies the corresponding abstract objects, and creates a world representation with instantiated object models. Similarly in task recognition, in the offline mode we prepare abstract task models on a computer that associate state transitions with the operations necessary to create such transitions. In the online mode, the system detects state transitions from the object recognition result and identifies an abstract task model to associate the detected state transition with an operation to achieve the transition.

The purpose of this task recognition is twofold. First, by dividing a continuous observation space into a discrete set of states and thus tasks, we can reduce the effect of observation errors. In the above example, the object recognition result contains a small positional error but because the four states are adequately discrete, the result is correctly classified as one of the four states. A few error-correction examples in the polyhedral world can be found in Suehiro and Ikeuchi (1992).

The second purpose is to separate observation from execution. The tasks obtained from observation are independent of the robot hardware; different robots can share the same observation module and only task-mapping modules need to be specific to the robot hardware. We can make robots with different hardware execute the same set of tasks by simply replacing the mapping module without changing the observation module.

Under this task-skill paradigm, we employ the divide-and-conquer strategy to find the appropriate task domains that have the necessary and sufficient task sets. These domains include two cubes (Ikeuchi et al. 1991), two polyhedral objects (Ikeuchi and Suehiro 1994), mechanical parts (Ikeuchi et al. 1993), and knotting rope world (Takamatsu et al. 2006). See Fig. 2.

In 2007, we demonstrated the application of this task-skill paradigm to a humanoid robot for performing a Japanese folk dance called Aizu-bandai-san (Nakaoka et al. 2007). We defined tasks for the lower body as contact states between the feet and the floor. The robot had three states: left-foot-contact, right-foot-contact, and both-feet-contact. For these states, we defined three task models: right step task, left step task, and standing task. For each task model, skill parameters were defined such as step width, step height, and waist height. See Fig. 3.

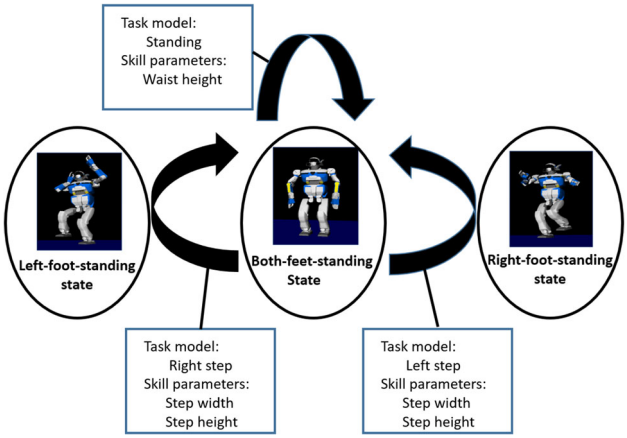

Fig. 3 Task models and skill parameters for lower-body motions. The robot has three states, left-foot-contact, right-foot-contact, and both-feet-contact. Based on these states, we define three task models: right step task, left step task, and standing task. For each task model, skill parameters are defined, such as step width, step height, and waist height

Although the robot could successfully perform the Aizu-bandai-san dance and attracted considerable attention from the media and academia, defining upper-body tasks (that is, describing human poses for this purpose) has been an open issue since then.

In this study, we designed tasks for upper-body operations (that is, motions of upper-body parts) based on Labanotation which is used by the dance community to describe dances. In addition, we propose a method to extract Labanotation by means of observation. The contributions in this paper are as follows:

1. Obtain Labanotation by means of observation,
2. Establishment of a method to describe upper-body operations as state transitions based on Labanotation, and
3. Generate upper-body operations of three actual robots based on observed Labanotation.

The remainder of this paper is organized as follows. Section 2 presents a review of related works. Section 3 explains Labanotation and its relationship with task models. Section 4 explains how to convert skeleton data from a depth sensor, such as a Microsoft Kinect, into Labanotation. Section 5 presents an overview of system implementation. Section 6 explains the demonstration results and evaluates the derived Labanotation. Section 7 concludes this paper.

2 Related Work

In robotics, many researchers have developed methods to adapt human motion to a humanoid robot. Riley et al. converted human motion data, obtained using a motion capture system, into joint trajectories of a humanoid robot to make the robot execute a dance routine (Riley et al. 2000). For the same purpose, Pollard proposed a method for constraining a set of given joint trajectories within the mechanical limitations of the joints (Pollard et al. 2002). For biped humanoid robots, Tamiya proposed a method that enabled a robot to follow given motion trajectories while maintaining balance (Tamiya et al. 1999). Kagami et al. extended the method to allow the robot to change the supporting leg (Kagami et al. 2000). Yamane proposed a dynamic filter that converts a physically inconsistent motion into a consistent one for a given body (Yamane and Nakamura 2003). These works were mainly concerned with creating a new joint trajectory within a given physical constraint. There has been no attempt to describe global motion structures using symbolic representations.

Most studies on humanoids have focused on generating lower-body motions because lower-body motions are critical for maintaining balance. Recently, however, humanoid robots are being used increasingly to perform day-to-day tasks, such as chatting. For those applications, generating appropriate upper-body motions are important.

In daily conversations, body language is an essential component. Such body language, a series of robot posters along with spoken language by the robot, conveys a feeling of liveliness to the users. For a robot such as Pepper, the lower body often consists of stable wheels, and the main issue is the generation of a meaningful set of upper-body posters or operations. LaViers and Egerstedt defined upper-body states and state transitions (LaViers and Egerstedt 2012). Furthermore, they represented emotional components and how such emotions influence robot operations. However, those states were extracted manually. By contrast, we aim to generate such upper-body motions based solely on observation.

The learning community is interested in the so-called "mimesis loop". One of their inspirations is the discovery of mirror neurons. A few representative works included (Vuga et al. 2013; Gams et al. 2015). In those works, observed motions are learned with reinforcement learning to mimic specific motions. Kawato's group proposed a humanoid robot to learn Okinawa-teodori based on the neural network approach (Cheng et al. 2008). The result is interesting but owing to the bottom-up nature of the learning mechanism, it is difficult to conduct a high-level symbolic analysis, such as an analysis of the dance structures learned. Moreover, no attempts have been made to extract symbolic representations that can be interpreted and edited by humans and transferred to robot hardware with different DOFs from the original hardware.

With regards to dance performance, Kuroki et al. enabled a biped humanoid to stably perform dance motions, including dynamic-style steps (Kuroki et al. 2003). Nakaoka et al. developed a similar dancing robot by using a software application called Choreonoid (Nakaoka et al. 2004). However,

these robots were coded manually, and no analysis is available. Kosuge et al. proposed a dance partner robot for western dances. The robot executed dance moves based on its partner's motion (Kosuge et al. 2003). Okuno's group developed a humanoid robot to step along with a musical beat (Yoshii et al. 2007). The motion of this robot was limited to stepping.

By contrast, in the present work we use Labanotation, a notation system used in the dance community (Guest 2005; Choensawat et al. 2010), to describe upper-body tasks for a humanoid robot. A few robotics researchers have proposed the use of Labanotation as the basis of a robot language design (Huang and Hudak 2003). We use Labanotation for describing upper-body states and designing tasks for upper-body operations of humanoid robots under the learning-from-observation paradigm.

The observation module of our system is related to human action recognition. Recently, there has been extensive research on the recognition of human actions based on visual observation. Representative works include the actionlet ensemble model (Wang et al. 2012), convolutional neural network (Ji et al. 2013; Chéron et al. 2015), trajectories (Wang and Schmid 2013; Amor et al. 2016), and motion characteristics (Jain et al. 2013; Kovashka and Grauman 2010). Several databases to evaluate the performance of recognition systems have been collated, including the UCF sports dataset (Rodriguez et al. 2008), Stanford Olympics dataset (Niebles et al. 2010), and Hollywood movie dataset (Marszalek et al. 2009). However, these methods are mainly concerned with the categorization of human actions such as biking, climbing stairs, and skipping. In fact, there is no notion of necessary and sufficient issues in those database and recognition. Rather, such recognition is limited to the original categorization of actions, necessitating a redesign of the recognition algorithms if the original categorization were modified even slightly. Moreover, such recognition results cannot be used to generate robot motion.

A few researchers have proposed the use of Laban movement analysis for gesture recognition and classification (Truong and Zaharia 2017; Rett and Dias 2007). This direction is important, and we use Labanotation to generate intermediate representations of our goal, which is to mimic human upper-body operations by using robots.

Bobick categorized the recognition of human actions into three classes: "movement", "activity", and "action" recognition. Among these three classes, movement recognition is closely related to our task recognition (Bobick 1997). Bobick defined a movement as "a motion whose execution is consistent and easily characterized by a finite space trajectory". We redefine a movement, a robot operation, as "a motion with a clear purpose to generate one state transition in one particular task domain". Our tasks are defined to specify corresponding movements for creating state transitions. We use the words motions, movements, and gestures interchangeably in this paper.

One of the imminent issues then is defining states and discerning such state transitions by observation. These states are characterized in various domains. In fact, we have been exploring this necessary and sufficient set of states in various domains of human operations, including the polyhedral world (Ikeuchi and Suehiro 1994) and lower-body dance motions (Nakaoka et al. 2007). Herein, we design states and state transitions for upper-body operations based on Labanotation.

3 Labanotation and Task Models

3.1 Introduction to Labanotation

A dance or a gesture consists of a series of human motions. Such a continuous sequence of motions can be described using Labanotation. Labanotation was developed by Rudolf V. Laban in the early 20th century as a method of movement notation (Guest 2005). Labanotation comprises four elements: body, space, time, and dynamics. Body stands for the body parts that move. Space represents motions described in terms of directions, levels, distance, or degree. Time represents the duration of a movement. Dynamics represent emotional components of motions. In this work, we focus on the first three basic elements (body, space, and time) and leave dynamics for future work.

A Labanotation score is drawn in two dimensions, body columns and time rows as shown in Fig. 4a. The vertical solid and dotted lines represent each body columns. Each column, corresponding to one body part, contains Labanotation symbols, such as rectangular and triangular symbols in Fig. 4a, representing how each body part moves along the flow of time. Time flows from the bottom to the top. Labanotation symbols are scaled to fit the starting time and the ending time, and the gap between two symbols in a column indicates a lack of motion in that period or holding the previous pose during that period.

The columns are divided into the left and the right sides, corresponding to the left and right sides of the body. These columns correspond to body parts and are depicted using the arrow-like symbols in Fig. 4b. The four support columns at the center represent the foot that supports the body. It is not always necessary to specify all the body parts. For example, because the focus of the present work is on upper-body motions, Fig. 4a only represents the upper arms, lower arms, and head explicitly. Other body parts such as left and right foot and their support information are omitted.

In Labanotation, symbol shapes, such as rectangular shape or triangle shape, represent the motion directions of body parts. The motion direction is specified using the coordi-

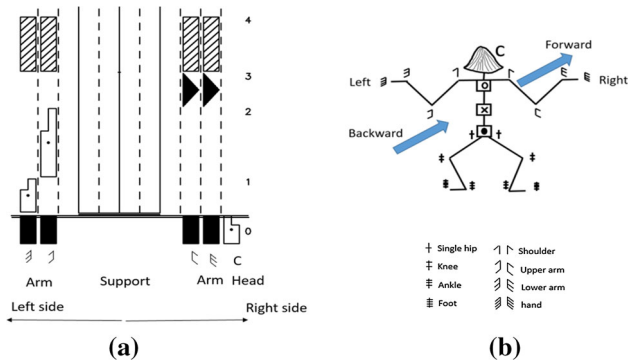

Fig. 4 Labanotation. **a** Labanotation score and **b** body part symbols corresponding to the columns in the score

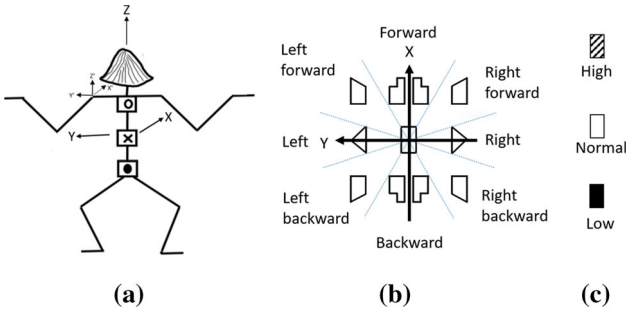

Fig. 5 Labanotation and its coordinate systems. **a** Body coordinate systems. The XYZ axes define the main body coordinate system, and local coordinate systems are defined for each body part. For example, X′Y′Z′ is the local coordinate system defined at the left upper arm, **b** shapes along azimuth directions and **c** shading along zenith angle

nate systems as shown in Fig. 5. For an upper-body motion, Labanotation defines the main body coordinate system, shown as the XYZ axis in Fig. 5a. The origin of the main body coordinate system (known as the "main cross" in the dance community) is at the center of the body when a dancer stands naturally. The X axis (front or back) is the direction in which the dancer is facing. The Y axis (left or right) is the direction from right to left. The Z axis (up or down direction) is the direction from the feet to the head.

For each body part, a local coordinate system, which is parallel to the body coordinate system, is defined at the near joint of that part to the body. The axis X′Y′Z′ in Fig. 5a is an example of the local coordinate system defined for the left upper arm.

Based on the local coordinate system, Labanotation defines 11 shapes for azimuth directions and three shadings for levels (or zenith directions). Azimuth directions include eight main directions, namely, forward (X axis), backward (− X axis), left (Y axis), right (− Y axis), left forward, right forward, left backward, right backward, and one special direction, the north–south axis (Z axis) referred to as "place". Figure 5b shows the corresponding eleven symbols.

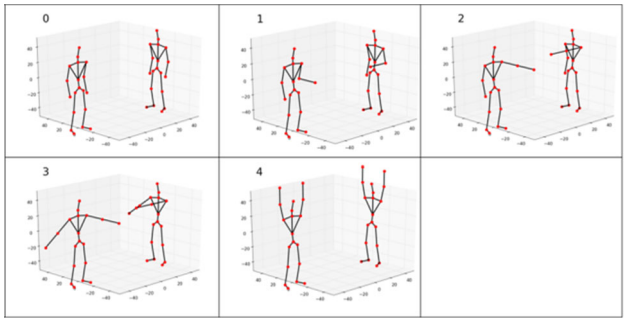

Fig. 6 Skeleton figures corresponding to the Labanotation score in Fig. 4a

Here the forward and the backward directions use two symbols each based on which body part points the forward and backward directions, respectively. Namely, for example, the left upper arm uses the left side symbol for the forward direction, and the right upper arm uses the right-side symbol for the forward direction. The levels (or zenith directions) include "high", "normal", and "low". Figure 5c shows the corresponding shadings.

Let us interpret the Labanotation score in Fig. 4a. Figure 6 shows an example of the end poses corresponding to the Labanotation score in Fig. 4a. At the initial pose, denoted by 0 in Fig. 4a, both arms point downward (the south pole), referred to as "place low". The head faces the forward direction. In task 1, which requires unit time, the left lower arm rises and finally points toward the front direction, referred to as "forward normal", as indicated in the skeleton figure in the 1st frame in Fig. 6. In task 2, which requires twice the unit time and thus the symbol in task 2 is twice longer than the symbol in task 1, the left upper arm is raised, and, as the result, the left upper and lower arms point toward the forward direction, while the right hand side keeps the same pose, which corresponds blank space in the Labanotation score. In task 3, keeping the left upper and lower arms in the same pose, the right upper and lower arms point toward the right low direction, referred to as "right low". In task 4, which requires twice the time as task 3, both arms point toward the upper direction, referred to as "place high".

3.2 Labanotation, Tasks, and States

Each symbol in a score carries two different types of information: the final pose and the duration. The final pose of each body part is represented by the shape and the shading of each symbol. The duration from the start to the end of the task is represented by the length of each symbol. Labanotation symbols do not describe the trajectory between the start and the end poses.

We can interpret one Labanotation symbol as indicating one robot task. One robot task is defined as one state transi-

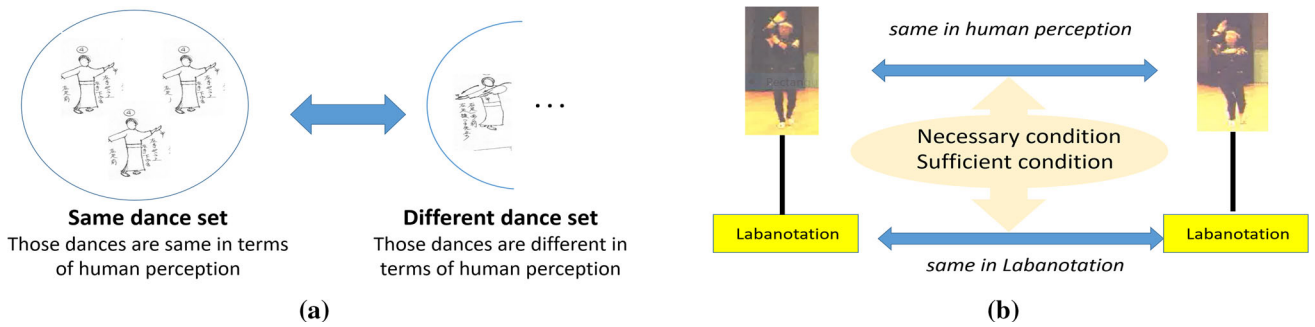

Fig. 7 Same dance set and Labanotation. **a** Same dance set and different dance set. Each small human figure depicts one dance performance. There is one group of different performances that are considered to be the same dance, while others are considered to be different dances and **b** Labanotation and human perception

tion in the learning-from-observation paradigm (Ikeuchi and Suehiro 1994). In this paper, we can define one state as one key pose of a robot. One Labanotation symbol corresponds to one state (pose) transition, of which the end state (pose) is depicted as the Labanotation symbol. The starting state corresponds to the previous pose depicted by the previous Labanotation symbol. The task duration is represented as the length of the symbol. One exception is the initial symbol below the double-line in Fig. 4b, which does not indicate any performance duration, but simply indicates the initial pose of the entire task sequence.

3.3 Labanotation as a Necessary and Sufficient Condition for Defining Equivalent Set of Dances

Labanotation scores not only visually resemble music scores but also have similar characteristics. A competent musician can record a given musical score by merely hearing it play. By reading a music score, a musician can play the same score with a slight variation in each performance. In a similar way, by watching a same dance performance, a dance expert can record its Labanotation score. By reading a given Labanotation score, a dancer can perform the same dance with a slight variation in each performance.

Equality of Labanotation scores is the necessary and sufficient condition for two dance performances to belong to an equivalent set of dances. Dances in general are not all considered the same. Yet, performances of a dance which differ slightly owing to the dancers performing them are considered the same dance by other dancers. We can define an equivalent dance set comprising the same aforementioned dance performances as small human symbols in Fig. 7a. Here, each small human symbol depicts one dance performance.

We can use Labanotation to check whether a dance performance belongs to one equivalent dance set. We can state that the necessary and sufficient condition for two dance performances to belong to the equivalent dance set is that the two dance performances can be described using the same Labanotation score. See Fig. 7b.

- *Sufficient condition* According to the Labanotation committee, if two dance performances belong to an equivalent dance set, they should be recorded using the same Labanotation score.
- *Necessary condition* By looking at the same Labanotation score, various dances will deliver various performances. However, the committee guarantee that these performances are perceived as the same dance by human dancers.

In fact, to tune this sense of the equivalent dance set, the Labanotation community has established a training course for recording dances, and a certificate is issued to those who pass this exam.

3.4 Justification of Labanotation

Even though digitization in terms of the directions and levels of Labanotation seems too coarse, it can be justified using psychological evidence. Labanotation samples only eight azimuth directions from 360-degree continuous directions, and five zenith directions from 180-degree continuous directions. Along this line of digitization, Miller's law declares that the human capacity for processing analog information into digital information allows for only seven, plus or minus two, categorizations (Miller 1956). For instance, the number of colors in a rainbow is seven, even though our eyes perceive continuous color variances in a rainbow. The number of main chords in music is exactly seven. When evaluating someone's performance, we often use five scales, namely, excellent, good, fair, bad, and worse. Based on this argument by Miller, the eight-directional digitization in terms of azimuth angle and the five-directional digitization in terms of zenith angle may be reasonable considering the limits of human perception, bolstered by the fact that the dance com-

munity has been using this notation for more than a century. Of course, in Labanotation, it is possible to specify finer directions if necessary, but to this end, we must define a sub-interval and probably around seven directions to digitize the sub-interval into finer directions.

Currently, many computer vision researchers focus on "recognition of gestures" to classify continuous human motions into categories such as "standing up" or "running". However, it is unclear why such categories are adequate to describe human motions. We propose an intermediate level of descriptions obtained from continuous human motions, digitization of timing, and part directions along the resolution based on the capacity of human perception by borrowing notations from the dance community. We believe that even though Labanotation is used by the dance community, we should be able to use it for describing general human motions. A Labanotation score, a series of Labanotation symbols, corresponds to each high-level recognition, such as "standing up" or "running" in the current gesture recognition scheme.

4 Labanotation Encoder

One of the main components in the observation module is the Labanotation encoder. The encoder, describing a motion sequence as a Labanotation score, involves two steps: key frame detection and Labanotation encoding. First, we must decide when an important pose occurs from a sequence of human motions. We must choose one point in time at which to record the pose using Labanotation symbols. We refer to such a point in time as a key frame. Then, we convert key poses at those key frames into Labanotation symbols.

4.1 Naïve Method for Key Frame Detection

One naïve method is to convert all poses of a human performer at each sampling time into Labanotation symbols without considering whether the same symbol was assigned to the previous step or not. The Labanotation transitions are then extracted.

We implemented this method. The number of key frames detected turned out to be dozens of times more than a Labanotation expert would record. This method cannot generate human-like Labanotation scores for three reasons. First, one motion may cross multiple areas in space. As an example, suppose that a body part moves from "place low" (south pole) to "place high" (north pole). The system generates multiple Labanotation symbols along the trajectory between the two poles. Second, for one single motion, different body parts start to move at slightly different times because of the nature of human performance. This leads to the creation of multiple key frames for a single meaningful motion. Third, from one pose to another, body part motions may hover around the detection threshold, creating noise and leading to the creation of extra Labanotation symbols.

This method performs poorly in generating meaningful Labanotation compared to human experts. However, the method does preserve intermediate trajectory information as symbolic representations. We can use this method for skill analysis in the future. In addition, we will use the results yielded by this method as ancillary information for the parallel energy method, which is described later.

4.2 Total Energy Method

Based on a discussion with Labanotation experts, brief stops in body motions play important roles in human perception and, thus, provide candidates for key frames. Shiratori et al. considered a motion energy function of all the components of the human body; that is, he combined the motion energy values of all motions of the hands, feet, and head, and determined key frames as local minima of the energy function (Shiratori et al. 2006). They assumed that key frames usually occur around beat points in the accompanying music. Key frames are searched for around beat points based on this assumption. Even though this idea of combining energy functions with the beat of music was powerful, it was limited in its applicability to applications other than dance analysis. Thus, we redesign the energy functions without considering music beats, based only on the speed and acceleration of the hands' positions.

Considering the noisy values and motion blur in the captured data, a smoothing process is first applied based on the discrete convolution of a Gaussian-based filter to the variances x, y, and z of each endpoint, (left and right hands in our case), separately:

$$f'(x) = f(x) * G(x), \tag{1}$$

with

$$\mathrm{G(x)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{2}$$

In our implementation, we use the sum of the energy of the left and the right hands. The basic idea of the total energy function for one side of the hand is as follows:

$$E = g(E_a(x, y, z)) - k(E_s(x, y, z)), \tag{3}$$

where E_a represents the acceleration calculated as

$$E_a(x, y, z) = \frac{1}{\sqrt{3}}\sqrt{\left(\frac{\partial^2 x}{\partial t^2}\right)^2 + \left(\frac{\partial^2 y}{\partial t^2}\right)^2 + \left(\frac{\partial^2 z}{\partial t^2}\right)^2}, \tag{4}$$

E_s is the speed determined as

$$E_s(x, y, z) = \frac{1}{\sqrt{3}}\sqrt{\left(\frac{\partial x}{\partial t}\right)^2 + \left(\frac{\partial y}{\partial t}\right)^2 + \left(\frac{\partial z}{\partial t}\right)^2}. \quad (5)$$

Here x, y, z is the hand position at t with respect to the body coordinate system. The energy terms, E_a and E_s, are normalized to have values in the interval [0, 1] based on the maximum and the minimum values over the entire observation. Namely,

$$\hat{E}_a = g(E_a) = \big(E_a - E_{a\min}\big)/\big(E_{a\max} - E_{a\min}\big), \quad (6)$$
$$\hat{E}_s = k(E_s) = \big(E_s - E_{s\min}\big)/\big(E_{s\max} - E_{s\min}\big), \quad (7)$$

The total energy is the summation of the left and the right-side energy:

$$E = \sum_{left,right} \widehat{E_a} - \sum_{left,right} \widehat{E_s}, \quad (8)$$

The total energy method detects peaks in the function and uses those corresponding frames as key frames.

4.3 Parallel Energy Method

Each body part performs its own task in a parallel manner. In Labanotation, each column may contain symbols of different lengths compared to those in other columns, as shown in Fig. 4a. For example, the left lower arm is raised to the middle height, while the other body parts hold the same positions as in the first key frame. This suggests giving each body part its own energy function.

We analyze the motions of each body part by using the spherical coordinate system because Labanotation is based on the spherical coordinate system. We can use the same local coordinate systems for each body parts as defined in the Labanotation. See Fig. 5a. Namely, we can set the origin of the local coordinate system at the joint near the body center, while the axes are aligned with the global body coordinate system. The joint position of the far side joint can be represented as (x, y, z) in this local coordinate systems. We convert these positions from the Cartesian coordinate system to the spherical coordinate system as follows:

$$r = \sqrt{x^2 + y^2 + z^2}, \quad (9)$$
$$\theta = \arccos\left(\frac{z}{r}\right), \theta \in [0, 180], \quad (10)$$
$$\phi = \arctan\left(\frac{y}{x}\right), \phi \in (-180, 180]. \quad (11)$$

We calculate the geodesic angular speed for all four upper-body parts (right upper and lower arms, left upper and lower arms) as follows:

$$\vec{v} = \big(\dot{\phi} sin\theta\big)\vec{e}_{\emptyset} - \dot{\theta}\vec{e}_{\theta}, \quad (12)$$

We define the energy function of each body part as its speed:

$$E_s = |\vec{v}|. \quad (13)$$

We employ two Gaussian filters with different σ to smooth the result of the energy function and to detect the precise positions of key frames.

$$E(x) = E_s(x) * G(x), \quad (14)$$

First, we apply the Gaussian filter with the larger σ to the energy function. We detect all local minimum points along the output. By given a rough position of local minimums on the larger scale, we update the position of the "true minimum" by searching in between two nearby inflection points on the energy function filtered by the Gaussian filter with the smaller σ. Here, we use the criteria $E''(x) = 0$ and $E'''(x) \neq 0$ to find out the inflection points of filtered energy function (Witkin 1984).

Holding detection Holding is the interval that a performer maintains the same pose for a while. An ideal shape of the energy function for a holding is as follows: a sudden drop in the valley, a valley shape of a certain width, followed by a rise. The longer the stop, the wider is the breath. However, for a human performer, it is impossible to keep a joint still. The value rises and falls around zero as the performer holds the position, which is represented by small peaks and valleys. We could set a small threshold to eliminate those small peaks and values for detecting the holding interval.

However, we do not want to set an arbitrary threshold. Instead, we use the naive labanotation generated by the naïve method. We consider that intervals as a holding when successive frames have the same naïve Labanotation with multiple valleys. We only keep the first valley and the last valley as the starting and ending frames, ignoring anything in between.

Brief stop detection After processing the holding frames, we examine remaining frames for brief stop detection. A brief stop can be detected at a point where the energy function has a valley in an interval of successive frames with the same naïve Labanotation.

Keyframe detection After collecting potential key frames of each body part separately, referred to as *part key frames*, we line up the result to find the real key frames, referred to as *body key frames*. As mentioned earlier, because of the limitation of human physical ability, to perform one set of

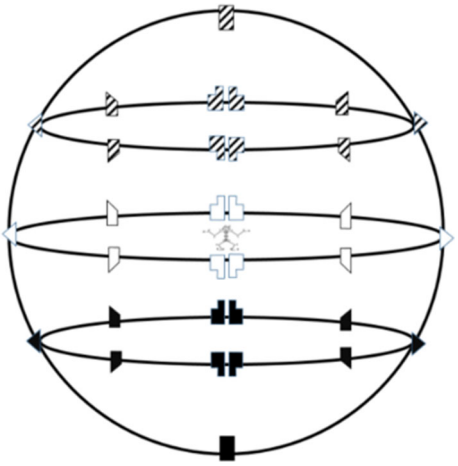

Fig. 8 26 Directions on Gaussian sphere. Each direction can be represented as a point on the Gaussian sphere. The distance between two directions can be measured as the geodesic distance between two corresponding spherical points

Labanotation, different body parts may reach their end positions at different timings. We assume that a body key frame is surrounded by several part key frames that belong to the same body key frame. Each body key frame is relatively far away from other body key frames. We allow the maximum time difference between two part key frames belonging to the same body key frame to be less than 1/3 s. We use a window of this size to scan along the timeline, clustering potential part key frames inside that window as one group, and then move to the next potential part key frame that has not been scanned. Then, we calculate the average position of part key frames within each group and consider those positions as the output body key frames of the parallel method.

4.4 Labanotation Encoding

The eight azimuth directions and five zenith directions Labanotation can be represented as 26 directions on the Gaussian sphere as shown in Fig. 8. Conversion of a pose in each key frame to Labanotation is done based on the angular direction of each part. The symbol corresponding to the nearest sampling direction among 26 directions measured by the geodesic distance is considered to be the corresponding notation.

The Labanotation committee defines assistant frames as intermediate frames along long trajectories. Suppose the angular difference of one part between two key frames is more than 135°. If the trajectory on the Gaussian sphere is along the geodesic of the sphere, the committee defines that no extra symbol is necessary. If not, an extra frame is inserted at the middle point for specifying the trajectory uniquely. We refer to this frame as an *assistant frame*. Because the generation of assistant frames relies strictly on the quality of key frames, we focus only on examining the efficiency of key frame generation by using different methods.

The Labanotation committee allows the use of two types of methods to specify the duration of a task: proportionate timing and free timing. In a proportionate timing method, the length of a symbol is a multiple of a unit length. Examples include task sequences of dance based on music beats. The other type of sequences consists of tasks with variable lengths, which are specified using the free timing method. For the sake of generality, we use the free timing method in this paper. Each Labanotation symbol has a length proportional to the actual frame number of two adjacent key frames.

5 System Implementation

5.1 System Overview

In this study, we focus on converting an input sequence into a Labanotation score. However, the purpose of the proposed system is to mimic human gestures through robot performance under the learning-from-observation paradigm. Thus, in this section, we explain the entire system that generates Labanotation scores in the observation module and directs robot performances based on the Labanotation scores in the task-mapping module.

Figure 9 shows an overview of the proposed system. In the observation module, a depth sensor is used to record human motions. From skeleton data obtained using the depth sensor, key frame extraction and Labanotation encoding, as described in Sect. 4, are conducted.

In the task-mapping module, Labanotation scores are converted into robot motions. A task-mapping module is specific to a robot configuration. The later part of this section explains the details of the module.

We used two custom-made and one commercially available robot, GR001 (G-robot 2016), for demonstrating the machine-independence of the observation module. One of the custom-made robots, RABOT1, shown in Fig. 10a, has seven DOFs, namely two DOFs around each shoulder, one DOF around the waist, and two DOFs for head motion (pitch and yaw). Each DOF of the robot is directly driven by a servo motor. The other custom-made robot, RABOT2, shown in Fig. 10b, has nine DOFs. This robot has one DOF for the body rotation, two DOFs for the head motion, and three DOFs for each of two arms. In addition, we used one commercially available robot, GR001 as shown in Fig. 10c. The upper-body configuration of GR001 is the same as that of RABOT2, although their arm lengths and control mechanisms are different.

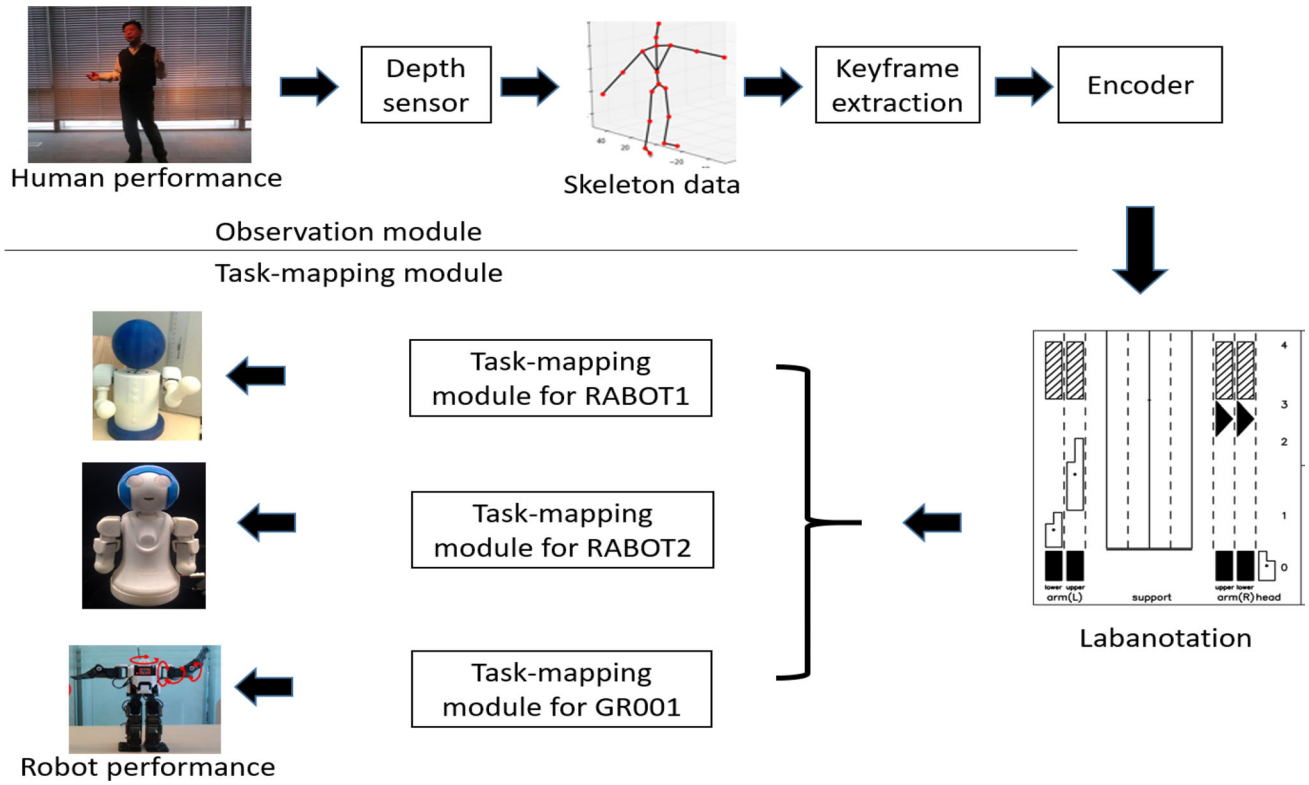

Fig. 9 Overview of the learning-from-observation system. The system consists of two modules: observation and mapping. The observation module records human motions using a depth sensor and then coverts those measurements into skeleton data. At the next step, the observation module extracts key frames using either the total (4.2) or the parallel (4.3) method. The skeleton data at the key frames are encoded into labanotation scores (4.4), which are independent on robot hardware. The task-mapping module converts Labanotation scores into robot motions for each specific robot hardware

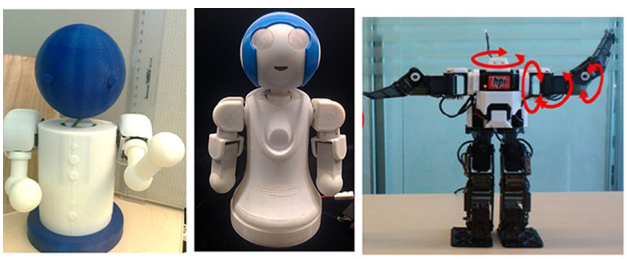

Fig. 10 Robot hardware employed in experiments. **a** 7DOFs RABOT1, **b** 9 DOFs RABOT2 and **c** GR001

5.2 Task-Mapping Module

A task-mapping module maps a Labanotation score to a sequence of motions of a robot. Each robot has different configurations; we prepared a mapping module corresponding to each robot. In this section, for the sake of clarity, first, we will explain the simple 7-DOF RABOT1 as a test bed.

In addition, we assumed that a Labanotation score represents hand parts in one column instead of three columns of upper arm, lower arm, and hand. Later, we will explain how to extend the method to other complicated cases. Some robots contain not only rotational joints but also prismatic joints or joints with different kinematic structures. However, for representing human motions, typical humanoid robots are the most suitable hardware. Thus, we limit our discussion to humanoid robots with rotational joints. The procedure for mapping Labanotation scores to robots with different types of joints will be discussed elsewhere.

As for the pitch direction of the whole arm, following Labanotation we digitized the direction into three levels: "high", "normal", and "low", as shown in Fig. 11a. The roll space of the whole arm is represented as a set of four positions, "left", "left forward", "forward", and "right forward", as shown in Fig. 11b. Owing to the limitations of RABOT1, only frontal poses are implemented. Of course, some human gestures may have more complicated motions, such as moving one arm backward. We ignore such motions in this simple implementation. All outside motions are represented as boundary motions. We found that such a simple implementation provides satisfactory results in most cases because human gestures are intended to be shown to other persons who are usually in front of the person performing

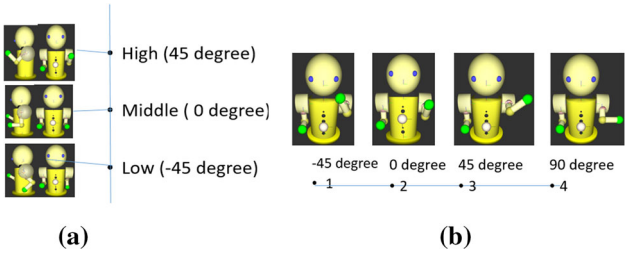

Fig. 11 Robot hardware employed in experiments. **a** Pitch and **b** roll

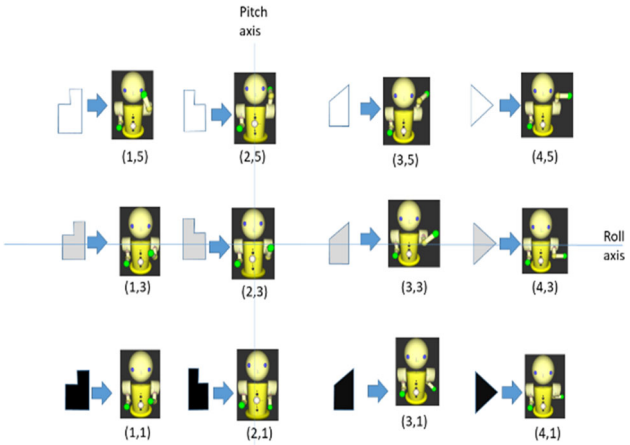

Fig. 12 Mapping Labanotation symbols to robot configurations

the gestures. All possible configurations of the left arm of RABOT1 are shown in Fig. 12. Then, we assign Labanotation symbols to those configurations.

In the above simple example, because the DOFs in Labanotation are identical to the DOFs of RABOT1, a simple mapping of poses of the body parts to Labanotation symbols works well. The similar mapping is possible on mapping Labanotation with upper and lower arm columns to the 9 DOFs RABOT2.

In a general case, we must consider two cases: a robot has more DOFs than a Labanotation score, and a Labanotation score has more DOFs than a robot.

Often, a robot has more DOFs than a Labanotation score owing to the limitation of the sensor used observe humans. A standard Labanotation score comprises three columns corresponding to the upper arm, lower arm, and hand, instead of one column as described above. However, owing to this limitation, we may have to omit some of the columns. Often, the lower arm and the hand columns are concatenated, as is the case for a Kinect sensor. When a robot has more DOFs than a Labanotation score, we simply map the concatenated directions to two robot parts. For example, a Labanotation symbol in the lower arm column is mapped both to the robot's lower arm and hand directions. We can apply the same idea to robots with higher complexity.

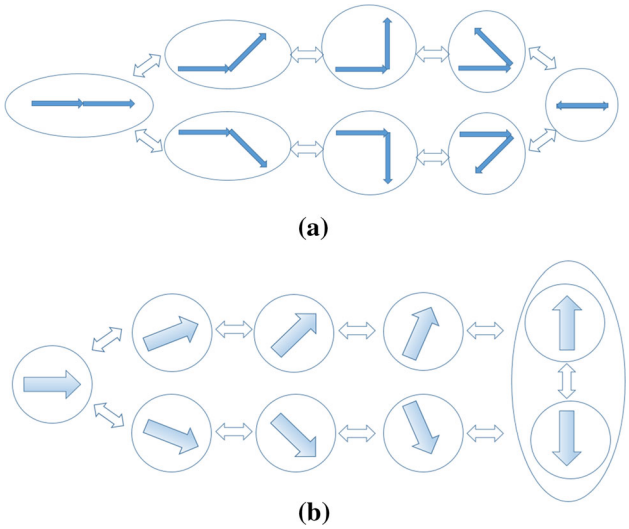

Fig. 13 Concatenation method. **a** Original configuration and **b** concatenated configuration

When a Labanotation score has more DOFs than a robot, we recursively combine adjacent Labanotation symbols into one symbol until the approximation is consistent with the robot's DOFs. Because each body part is connected to each other and Labanotation digitizes the direction in steps of 45°, possible configurations between two parts consist of eight cases: continue, forward diagonal, orthogonal, backward diagonal, and reverse, as shown in Fig. 13a. Then, the reachable directions are eight, as shown in Fig. 13b. One singular case occurs in the reverse position. By considering the history of transition, either direction is selected. These eight cases are true for directions and levels. This concatenation is mapped to the directions of robot parts. Fortunately, Labanotation denotes each body part separately. The necessary depth of the recursive operation to be considered is three for upper-body motion.

5.3 Trajectory Generation

A task only provides the start and end poses represented by Labanotation symbols. For a robot motion, we need a trajectory to specify the intermediate motions between the two poses. In this study, we implemented a linear interpolation method for determining joint angles.

5.4 Evaluation of System Performance

We compared human motions with robot motions between two key frames, as shown in Fig. 14. The top and bottom rows show the original human and robot motions, respectively. The poses surrounded by the dotted boxes are the poses in key frames. Because the motion speeds of the three robots are different owing to their different control mechanisms, the

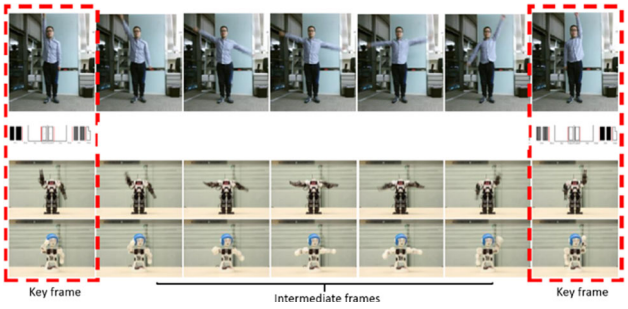

Fig. 14 Poses in key frames and intermediate frames. The first row is a sequence of human performance, second row shows Labanotation scores corresponding to key poses, and the third and fourth rows show robot performance

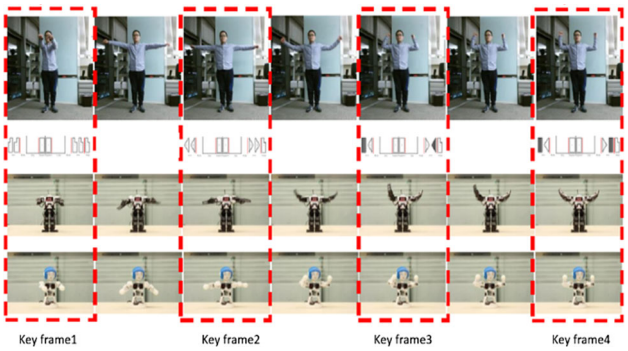

Fig. 15 Key poses

intermediate poses captured between the two key poses are different. However, from visual inspection, human observers tend to neglect such differences in intermediate poses. They only care about the key poses in the key frames, as expected. In fact, this finding supports our argument about the necessity and sufficiency of Labanotation for the equivalent dance set.

Figure 15 shows the key poses in the key frames. Based on visual inspection, again, we can consider that the system can reproduce the original motion reasonably well.

6 Evaluation of Labanotation Encoder

6.1 Test Data

We evaluated the performance of the Labanotation encoder. We tested our encoding methods on the following six daily conversation motions:

- Sequence 1: waving right arm (Fig. 16a),
- Sequence 2: moving arms up and down like a weighing scale (Fig. 16b),
- Sequence 3: drawing three horizontal lines at different heights (Fig. 16c),

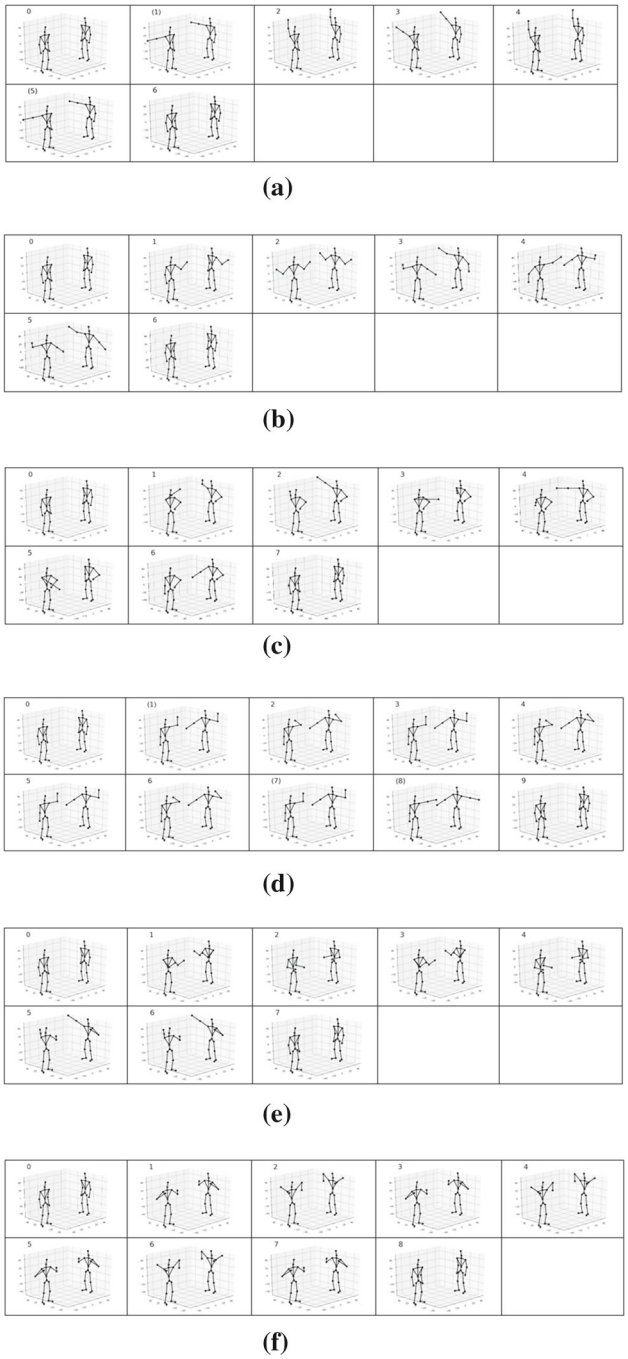

Fig. 16 Six daily conversation motions as test data. **a** Waving right arm, **b** moving arms up and down like a weighing scale, **c** drawing three horizontal lines at different heights, **d** one arm pointing at a place, waving the other arm to ask someone to come here, **e** moving hands up and down according to some rhythm and pointing one hand at a specific place and **f** chicken wing dance

- Sequence 4: one arm pointing at a place, waving the other arm to ask someone to come there (Fig. 16d),
- Sequence 5: moving hands up and down according to some rhythm and then pointing one hand at a specific place (Fig. 16e),

Table 1 Key frame detection

Loss of key frames	Sq1	Sq2	Sq3	Sq4	Sq5	Sq6
Total	0	0	0	0	1	1
Parallel	0	0	0	0	0	0

- Sequence 6: Chicken wing dance (Fig. 16f).

We asked a Labanotation expert, who has a license from the Labanotation committee, to generate Labanotation. This license system guarantees the consistency of the generated Labanotation, independent on individual Labanotation experts. The serial number of an assistant frame is denoted by parenthesis in Fig. 16.

6.2 Key Frame Detection

We evaluated the performance of the total and the parallel methods by using the aforementioned six sequences. Table 1 shows the comparisons of key frame detection. In this comparison, the assistant frames are not used to calculate the error between the Labanotation generated automatically and that generated by the Labanotation expert. The total energy method failed to extract one key frame each from Sequences 5 and 6, while the parallel method detected all key frames successfully.

6.3 Distance Among Labanotation

We compared our encoder-generated Labanotation scores from the extracted key frames with those generated by the Labanotation expert. To evaluate the quality of these Labanotation scores, we defined the distance measure between two Labanotation scores. A Labanotation score of one sequence, L, can be represented as a matrix, the columns of which correspond to each of the body parts and the rows correspond to each of the key frames. Each element of the matrix represents the body direction, which is digitized into 26 directions, each corresponding to one Labanotation symbol, as shown in Fig. 8. Each component of the matrix represents the body direction digitized into 26 directions. Since we limited our scope to the upper body in the present study, given T sampling along the time axis, each dance can be represented as a $T \times 4$ matrix. Here, the subscripts l–w, l–e, r–e, and r–w denote the left lower arm, left upper arm, right upper arm, right lower arm, respectively.

$$L = \begin{bmatrix} l_{\{t,l-w\}} & l_{\{t,l-e\}} & l_{\{t,r-e\}} & l_{\{t,r-w\}} \\ \vdots & \vdots & \vdots & \vdots \\ l_{\{2,l-w\}} & l_{\{2,l-e\}} & l_{\{2,r-e\}} & l_{\{2,r-w\}} \\ l_{\{1,l-w\}} & l_{\{1,l-e\}} & l_{\{1,r-e\}} & l_{\{1,r-w\}} \end{bmatrix} \tag{15}$$

Table 2 Distance among generated Labanotation

Error	Sq1	Sq2	Sq3	Sq4	Sq5	Sq6
Total	8.75	8.55	5.07	5.56	9.49	10.93
Parallel	8.00	8.55	5.07	4.81	6.30	7.62

Each spatial direction can be represented as a point on the Gaussian sphere. The difference measure between two directions can be defined as the geodesic distance between the two corresponding points over the Gaussian sphere.

We can define the distance measure between two Labanotation scores, L and M, as the summation of the geodesic distances of each element:

$$D = \sum_{i=1}^{T} \sum_{j=1}^{4} d(l_{ij}, m_{ij}) \tag{16}$$

Here $d(a, b)$ is the geodesic distance over the Gaussian sphere of two spherical points a and b.

Suppose we have two matrices A and B. A is extracted by a human expert, while B is extracted by a program. A and B may have different number of rows. First, we pair the rows in B with the closest counterparts in A. Some rows might not be paired. Two cases are possible:

1. B has rows that cannot be paired with A. This indicates that the program extracted more key frames than the human expert. Those extra rows are ignored from the calculation because there is no negative effect of inserting extra frames and key poses on robot performance.
2. B has less rows than A. This indicates that the program missed the necessary key frames extracted by the human expert. We penalize each deficient row with the worst case that may occur, which means all four body parts are in opposite directions (Table 2).

In terms of the distance of the generated Labanotation scores, errors generated by the parallel method are less than those generated by the total method.

7 Conclusion

In this paper, we proposed the use of Labanotation as the basis for defining robot tasks under the learning-from-observation paradigm in the domain of upper-body motions. We constructed a robot system to observe and mimic upper-body motions of humans. By observing human motions, we first extracted key frames, where one part of a human body briefly stops, by analyzing upper-body motions. To accomplish hardware independency, we introduced Labanotation as

the basic method for task representation. Because Labanotation is independent of robot hardware, tasks modeled based on Labanotation can be executed on different robot platforms by simply replacing mapping modules. We implemented the proposed system on two custom-made robots as well as a GR001 robot.

By using Labanotation, continuous motions can be effectively compressed and encoded. This compression is essential for a cloud robot, which is connected to a cloud computer and receives a few motion commands through a narrow channel between the robot and a cloud computer. In Labanotation, the entire motion space is divided into specific symbols, so that any gesture can be classified into a combination of symbols with a reasonable degree of coarseness corresponding to human perception.

For ensuring system completeness, we included an initial implementation of a task-mapping module for Labanotation, which controls robots on what-do-to. The module depends on the configuration of each robot as well as its control mechanism. Further discussion is necessary to consider the relationship between Labanotation and robot configurations. However, such a discussion is beyond the scope of computer vision studies.

Another important aspect of the mapping module is skill. In this study, we do not focus on skills, namely, trajectory variance. We simply interpolate the intermediate motions by linearly interpolating the joint angles. Laban proposed the Laban effort graph to characterize trajectories and speed variances along a trajectory by using symbolic representations such as Sudden, Smooth, Direct, and Indirect. In the future, we will implement this trajectory specification based on observation and characterization based on Laban's efforts. It is also necessary to relate the effort graph with to human motions.

We used the upper and lower body to generate different representations. This is because the upper-body motions can be designed with little concern about dynamic balance for realistic representation of such motions, while the design lower-body motions strictly require consideration of dynamic balancing to avoiding falls. As for the lower-body motions, Nakaoka et al. (2007) covers the topic.

In this study, we employed analytic functions to detect key frames. In addition, we used a 3D sensor to extract stick figures from the input sequences. It is interesting if we can obtain such information from 2D video streams by using DNN techniques. We will continue such efforts in the future.

Currently, many computer vision researchers focus on "recognition of gestures" to classify continuous human motions into categories such as "standing up" or "running". It is unclear whether such categorizations provide the necessary and sufficient classes to describe human perception and how each category, such as given by standard databases, is related with each other. We can use Labanotation as intermediate representations that describe such motions and guarantee equivalence in classes of human perceptions as described in Sect. 3.4. One of the interesting applications is to use Labanotation scores for categorization of action classes, such as "standing up" or "running" and to find common sets of symbol combinations that infer the underlying meanings of each action. For example, which part of the observed motions does really mean "running?" We can also provide a taxonomy of actions based on the distance measures of Labanotation scores.

Acknowledgements A part of this work was supported by Microsoft Research Asia Core project 2016 and 2017. Yoshihiro Sato, Naohiro Hayashi, Masaaki Fukumoto, and Ambrosio Blanco helped us to set up the experimental environment including two home-made robots, RABOT1 and RABOT2. Discussions with David Baumert, Yutaka Suzue and John Lee were also valuable. The authors acknowledge and appreciate their help.

References

Aboshosha, A., & Zell, A. (2003). Adaptation of rescue robot behaviour in unknown terrains based on stochastic and fuzzy logic approaches. In *Proceedings 2003 IEEE/RSJ international conference on intelligent robots and systems* (IROS) (Vol. 3, pp. 2859–2864).

Amor, B. B., Su, J., & Srivastava, A. (2016). Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 38*(1), 1–13.

Bobick, A. F. (1997). Movement, activity and action: The role of knowledge in the perception of motion. *Philosophical Transactions: Biological Sciences, 352*(1358), 1257–1265.

Cheng, G., Hyon, S.-H., Ude, A., Morimoto, J., Hale, J. G., Hart, J., Nakanishi, J., Bentivegana, D., Hodgins, J., Atkienson, C., Mistry, M., Schaal, S., & Kawato, M. (2008). CB: Exploring neuroscience with a humanoid research platform. In *2008 IEEE international conference on robotics and automation* (ICRA) (pp. 1772–1773).

Chéron, G., Laptev, I., & Schmid, C. (2015). P-CNN: Pose-based CNN features for action recognition. In *2015 IEEE international conference on computer vision* (ICCV) (pp. 3218–3226).

Choensawat, W., Takahashi, S., Nakamura, M., Choi, W., & Hachimura, K. (2010). Description and reproduction of stylized traditional dance body motion by using Labanotation. *Transactions of the Virtual Reality Society of Japan (VRSJ), 15*(3), 379–388.

Dombre, E., Duchemin, G., Poignet, P., & Pierrot, F. (2003). Dermarob: A safe robot for reconstructive surgery. *IEEE Transactions on Robotics and Automation (IEEE TRA), 19*(5), 876–884.

Gams, A., Kieboom, J., Dzeladini, F., Ude, A., & Ijspeert, A. (2015). Real-time full body motion imitation on the COMAN humanoid robot. *Robotica, 33*(5), 1049–1061.

G-ROBOTS. (2016). http://www.hpirobot.jp. Accessed 6 January 2016.

Guest, A. H. (2005). *Labanotation the system of analyzing and recording movement* (4th ed.). New York: Routledge.

Huang, L., & Hudak, P. (2003). Dance: A declarative language for the control of humanoid robots. *Research Report No. Yaleu/DCS/RR-1253*. Yale University.

Ikeuchi, K., Kawade, M., & Suehiro, T. (1993). Assembly task recognition with planar, curved and mechanical contacts. In *Proceedings IEEE international conference on robotics and automation* (ICRA) (Vol. 2, pp. 688–694).

Ikeuchi, K., & Suehiro, T. (1994). Toward an assembly plan from observation I Task recognition with polyhedral objects. *IEEE Transactions on Robotics and Automation (TRA), 10*(3), 368–385.

Ikeuchi, K., Suehiro, T., Tanguy, P., & Wheeler, M. D. (1991). Assembly plan from observation. In *Annual research review* (pp. 37–53). The Robotics Institue, Canegie Mellon University.

Jain, M., Jégou, H., & Bouthemy, P. (2013). Better exploiting motion for better action recognition. In *2013 IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 2555–2562).

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 35*(1), 221–231.

Kagami, S., Kanehiro, F., Tamiya, Y., Inaba, M., & Inoue, H. (2000). Auto-balancer: An online dynamic balance compensation scheme for humanoid robots. In *Fourth international workshop on algorithmic foundation of robotics*.

Kosuge, K., Hayashi, T., Hirata, Y., & Tobiyama, R. (2003). Dance partner robot—Ms DanceR. In *Proceedings 2003 IEEE/RSJ international conference on intelligent robots and systems* (IROS) (Vol. 4, pp. 3459–3464).

Kovashka, A., & Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition* (CVPR) (pp. 2046–2053).

Kuroki, Y., Fujita, M., Ishida, T., Nagasaka, K., & Yamaguchi, J. (2003). A small biped entertainment robot exploring attractive applications. In *IEEE international conference on robotics and automation* (ICRA) (Vol. 1, pp. 471–476).

LaViers, A., & Egerstedt, M. (2012). Style based robotic motion. In *American control conference* (ACC) (pp. 4327–4332).

Marszalek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 2929–2936).

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review, 63*(2), 81–97.

Nakaoka, S., Nakazawa, A., & Ikeuchi, K. (2004). An efficient method for composing whole body motions of a humanoid robot. In *Proceedings of the tenth international conference on virtual systems and multimedia* (VSMM2004) (pp. 1142–1152).

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., Hirukawa, H., et al. (2007). Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *The International Journal of Robotics Research, 26*(8), 829–844.

Niebles, J. C., Chen, C.-W., & Li, F.-F. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision* (ECCV) (pp. 392–405). Berlin: Springer.

Pollard, N. S., Hodgins, J. K., Riley, M. J., & Atkeson, C. G. (2002). Adapting human motion for the control of a humanoid robot. In *Proceedings 2002 IEEE international conference on robotics and automation* (ICRA) (Vol. 2, pp. 1390–1397).

Rett, J., & Dias, J. (2007). Human–robot interface with anticipatory characteristics based on Laban Movement Analysis and Bayesian models. In *2007 IEEE 10th international conference on rehabilitation robotics* (ICORR) (pp. 257–268).

Riley, M., Ude, A., & G. Atkeson, C. (2000). Methods for motion generation and interaction with a humanoid robot. In *AAAI and CMU workshop on interactive robotics and entertainment*.

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In *2008 IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 1–8).

Roy, A., Krebs, H. I., Williams, D. J., Bever, C. T., Forrester, L. W., Macko, R. M., et al. (2009). Robot-aided neurorehabilitation: A novel robot for ankle rehabilitation. *IEEE Transactions on Robotics (TRO), 25*(3), 569–582.

Shiratori, T., Nakazawa, A., & Ikeuchi, K. (2006). Synthesizing dance performance using musical and motion features. In *Proceedings 2006 IEEE international conference on robotics and automation* (ICRA) (pp. 3654–3659).

Suehiro, T., & Ikeuchi, K. (1992). Towards an assembly plan from observation: Part II: Correction of motion parameters based on fact contact constraints. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (IROS) (Vol. 3, pp. 2095–2102).

Takamatsu, J., Morita, T., Ogawara, K., Kimura, H., & Ikeuchi, K. (2006). Representation for knot-tying tasks. *IEEE Transactions on Robotics (TRO), 22*(1), 65–78.

Tamiya, Y., Inaba, M., & Inoue, H. (1999). Realtime balance compensation for dynamic motion of full-body humanoid standing on one leg. *Journal of the Robotics Society of Japan, 17*(2), 268–274.

Treptow, A., Cielniak, G., & Duckett, T. (2005). Active people recognition using thermal and grey images on a mobile security robot. In *Proceedings 2005 IEEE/RSJ international conference on intelligent robots and systems* (IROS) (pp. 2103–2108).

Truong, A., & Zaharia, T. (2017). Laban movement analysis and hidden Markov models for dynamic 3D gesture recognition. *EURASIP Journal on Image and Video Processing, 2017*(1), 52.

Vuga, R., Ogrinc, M., Gams, A., Petrič, T., Sugimoto, N., Ude, A., & Morimoto, J. (2013). Motion capture and reinforcement learning of dynamically stable humanoid movement primitives. In *IEEE international conference on robotics and automation* (ICRA) (pp. 5284–5290).

Wada, K., & Shibata, T. (2007). Living with seal robots—Its sociopsychological and physiological influences on the elderly at a care house. *IEEE Transactions on Robotics (TRO), 23*(5), 972–980.

Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 1290–1297).

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE international conference on computer vision* (ICCV) (pp. 3551–3558).

Witkin, A. (1984). Scale-space filtering: A new approach to multi-scale description. In *IEEE international conference on acoustics, speech, and signal processing* (ICASSP) (Vol. 9, pp. 150–153).

Yamane, K., & Nakamura, Y. (2003). Dynamics filter—concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation (TRA), 19*(3), 421–432.

Yoshii, K., Nakadai, K., Torii, T., Hasegawa, Y., Tsujino, H., Komatani, K., Ogata, T., & Okuno, H. G. (2007). A biped robot that keeps steps in time with musical beats while listening to music with its own ears. In *IEEE/RSJ international conference on intelligent robots and systems* (IROS) (pp. 1743–1750).