NAIST-IS-DD1061010

Doctoral Dissertation

Construction of Reconfigurable Motion Database for Real-Time Human-Robot Interaction

Yutaka Kondo

March 15, 2013

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Yutaka Kondo

Thesis Committee:

Professor Tsukasa Ogasawara	(Supervisor)
Professor Kiyohiro Shikano	(Co-supervisor)
Associate Professor Jun Takamatsu	(Co-supervisor)
Assistant Professor Kentaro Takemura	(Co-supervisor)

Construction of Reconfigurable Motion Database for Real-Time Human-Robot Interaction[*]

Yutaka Kondo

Abstract

Today, realization of communications with human and robot is crucial for actively researched human-robot interaction (HRI) and the future direction of human-robot symbiosis. Especially, natural body gesture for visual information, as well as speech dialog for audio information, is very important for human-like robots (*e.g.,* humanoids and androids). The visual information is commonly used in all countries to express muted feeling or intendment. In addition, for androids which have human-like appearance, human-likeness in the gestures is strongly required.

Therefore, in this thesis, we propose a method to generate android's body gesture by real-time reconfiguring a large-scaled motion database which captured humans' motions. This method realizes an autonomous HRI system which can interact reactively and naturally based on a human's behavior and response.

First, we propose methods for the construction of the reconfigurable motion database and the real-time generation of gestures. Then, we describe fundamental mechanism of the methods. To construct the database, one of natural language processings, a *Bag-of-words* which aims at similar sentence classification, is applied into a similar motion classification. Since the motion sequences are described in a frequency domain, these sequences can be classified rapidly based on the semantic similarity which the motion has. After a dynamic programming matching to more deeply classify the similar motions, the motions (*e.g.,* gestures) are manually given an appropriate parameter (*i.e.,* the target hand position and facial direction in pointing gesture's case). The gesture is appropriately synthesized to satisfy a current HRI situation. In addition, we

[*] Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1061010, March 15, 2013.

propose a motion connection method which enables to smoothly and rapidly react an user-intended interruption during interaction.

Finally, the proposed method is implemented on an android, *Actroid-SIT* and we develop a novel HRI system which can communicate with multiple people. Over 1,700 subjects in total were taken part in evaluation experiments. We confirmed that there are significant differences among the proposed system and conventional systems for every evaluation items, the response/residence time of communication, speaker ratio to the Actroid, and the impression of the Actroid.

Keywords:

body gesture, motion database, real-time planning, human-likeness, android,
human-robot interaction

実時間ヒューマン・ロボットインタラクションのための
再構成可能な動作データベースの構築*

近藤 豊

内容梗概

現在盛んに研究が行われているヒューマン・ロボットインタラクション（HRI）や，将来始まる人ロボット共生社会において，人・ロボット間の対話の実現は必要不可欠である．特に，ヒューマノイドやアンドロイドなどのような人型ロボットの場合，聴覚情報である音声会話の生成に加え，視覚情報であるボディジェスチャの生成も重要課題となる．ジェスチャは身体言語とも表されるように，音声言語だけでは伝えることできない感情や意図などの伝達手段として広く用いられている．さらに，アンドロイドはその人間に似た外見的特徴から，ジェスチャにおいても人間に似た動作表現が要求される．

そこで，本研究では，人間の動作を計測することで得られた大規模な動作データベースを実時間で再構成することで，アンドロイドにおけるボディジェスチャを計画する．これにより，人間の行動・反応に基づいて，応答性が高く自然なインタラクションが可能な自律HRIシステムが実現される．

初めに，提案手法である再構成可能な動作データベースの構築方法と，ジェスチャの実時間生成を提案し，その基本的な原理について述べる．データベースの構築には，自然言語処理の類似文章分類手法の1つであるBag-of-wordsを，類似動作分類手法として応用する．このとき，動作軌跡を周波数空間で扱うことにより，ジェスチャ自身が持つ意味的な類似尺度を基に，高速に分類することが可能となる．得られた類似動作は，動的計画法により，より詳細に分類が行われ，各類似動作群，つまり各ジェスチャには適当なパラメータ（例えば，手差し動作の場合，目標手差し位置と顔の方向）が与えられる．このパラメータを基に，現在のインタラクションに適した動作が生成されるよう類似動作同士が合成される．

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文，NAIST-IS-DD1061010, 2013年3月15日．

また，インタラクション実行中のスムーズかつ安全な割り込み手法を合わせて提案することにより，相手の対話に対して瞬時に応答を返すことが可能となる．

最後に，提案手法をアンドロイド Actroid-SIT に実装し，多人数とのインタラクションが可能な自律 HRI システムを開発する．そして，その設計仕様の詳細を述べるとともに，提案システムを用いた被験者実験を行う．累計 1,700 名以上が参加した実験結果から，従来システムと比べ，応答時間，対話時間，ロボットへの対話開始率，ロボットの印象などすべての評価項目において，有意な差が確認された．

キーワード

ボディジェスチャ, 動作データベース, 実時間プランニング, 人間らしさ,
アンドロイド, ヒューマン・ロボットインタラクション

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

To respond declining birth rate and aging population, robots are expected to replace a part of human's workforce. Recent robots require not only accuracy and durability, which industrial robots have, but also communication and entertainment for capabilities of human-robot interaction (HRI) and human-robot symbiosis. In HRI, realizing communication between a human and a robot is crucial [1]. And agents such as robots or computer graphing (CG) agents have been attracting attention as presenters for explanation and guidance to humans [2].

When communicating with human, visual information (*e.g.,* body gesture and facial expression) is very important factor as well as audio information (*e.g.,* speech dialogue). The visual information is commonly used in all countries to express muted feelings [3]. In addition, robots are superior to CG agents as presenters, because they can indicate a real-world object by pointing or gazing [4]. Especially for androids which have human-like appearance, human-likeness and interactivity in the gestures are strongly required to avoid the *Uncanny Valley* [5, 6].

On the other hand, Google CEO Eric Schmidt advocated that *Cloud Computing* technologies has been rapidly developed these years [1]. The idea of the cloud computing is being expanded into not only computers but also every electronics such as smartphones, home electronics, tiny gadgets, and sensor network. Robots are ones of the electronics. Recently, the cloud networked robotics has also been proposed by several researchers [7, 8, 9]. In the cloud networked robotics, a platform layer locates between service applications and robotic components. It isolates and coordinates them to realize multi-area, multi-robot networked robotic services. This fact suggests that robots assist humans through real-time communication based on huge computational power and database.

[1] http://www.google.com/press/podium/ses2006.html

1.2 Research Aim and Approach

1.2.1 Research Aim

There are two approaches for realizing human-like gesture expressions. The first approach is an online motion planning approach which is based on a empirical model. Flash *et al.* [10] proposed a minimum jerk (*i.e.*, the differential of acceleration) trajectory of human arm's reaching movement. Chikaraishi *et al.* [11] proposed an attractor selection model of natural idling movement.

The second approach is a database reutilization of human's motions captured by a motion capture system. Recently, we can easily capture human's motion, because optical or image based motion capturing systems are common nowadays. *Motion Graph* [12] generates motions by smoothly connecting with similar poses between motions. Inamura *et al.* [13] use a *Hidden Markov Model* [14] to blend different motions.

Since it is difficult to design empirical models for all gestures which have a wide variety of motion expressions, we propose a novel approach which integrates online planning methods and data-driven methods to fulfill aforementioned requirements; human-likeness and interactivity. This proposed motion planning method, named a *Reconfigurable Motion Database* (RMDB) has following three main features.

- **Motion classification**
  Collect semantic similar motions captured by different motion capture systems to construct a pre-designed motion database.

- **Motion parameterization**
  Synthesize similar motions according to motion's parameter which is fulfilled user demands such as a speaker and/or on object location.

- **Motion interruption**
  Transition motion to motion anytime smoothly and safely by online motion planning to rapidly react user-intended interruption in HRI.

First, RMDB constructs a pre-designed motion database to classify similar motions. The motions are described in a frequency domain and transforms them into features, *Bag-of-motion-features* which is inspired by a sentence retrieval method, *Bag-of-words* [15] and an image retrieval method, *Bag-of-features* [16]. *Bag-of-motion-features* can collect similar motions even captured by different motion capture systems as shown in Figure 1.1 (a).

Then, RMDB gives a parameter for each similar motion cluster (*e.g.,* gesture). The parameter indicates target location of a hand, target direction of a head, or target breadth of both hands. These similar motions are synthesized based on the parameter by *Match Web* [17] for generating target gesture.

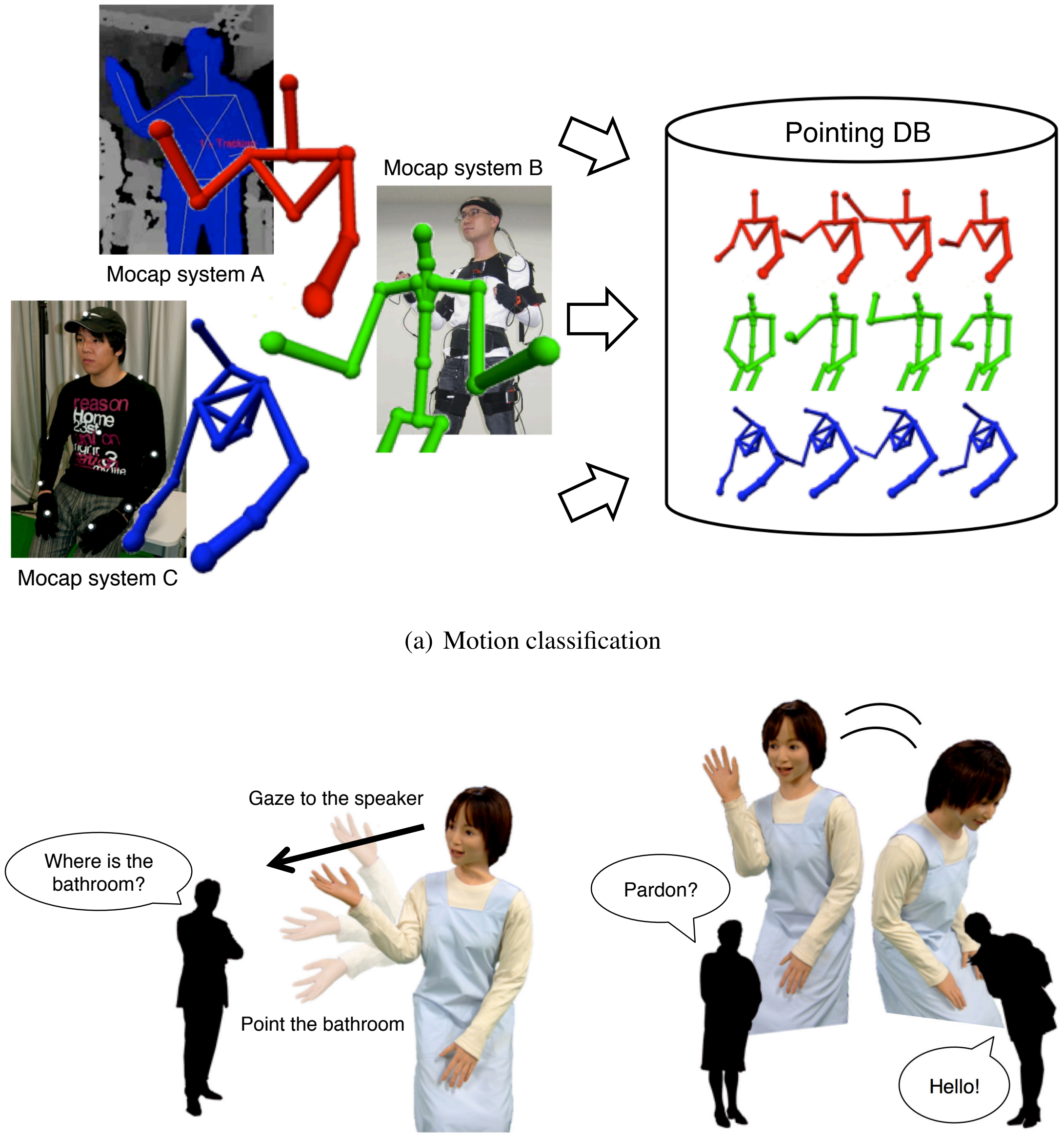
In HRI, the interaction should be enabled to rapidly react user-intended interruption such as topic/speaker switching and speech recognition failure. RMDB can transition motion to motion smoothly without self-collision inspired by a motion connection method, *Motion Graph* [12] and an online planning method, *Probabilistic Roadmap Method* (PRM) [18].

In addition, most previous research in HRI only dealt with communication between one robot and one person. Communication is not only with one-to-one but also with multi-party. Although several research methods tackled multi-party communication, these methods focused mainly on the recognition of multi-party conversations (*e.g.,* [19, 20, 21]).

1.2.2 Approach

We therefore develop an android system which enables multi-party communication using the RMDB. We also considered natural gaze movement, such as the convergence of both eyes, and a ratio of eye angle to head angle, based on the knowledge of human and chimpanzee [22, 23, 24]. The system can communicate smoothly with multiple people by rapidly switching to each person, and adjusting a gesture to the location of speaker and/or object as shown in Figure 1.2. These functions are also applied even while one-to-one HRI, since one-to-one HRI is the subset of multi-party HRI. The number of situations which our system can deal with are exponentially increased due to combinatorial explosion compared to conventional one-to-one HRI systems. We solve this problem by cooperation between a *Key-Value Store*, which has ACID (Atomicity, Consistency, Isolation, and Durability) properties, and other system components. Therefore, we do not have to be concerned about the synchronization and scalability of the system.

To evaluate the effectiveness of our proposed system, we conduct a lot of subject



(a) Motion classification

(b) Motion parameterization

(c) Motion interruption

Figure 1.1: Three main features of the *Reconfigurable Motion Database*: (a) Motion classification, (b) Motion parameterization, and (c) Motion interruption

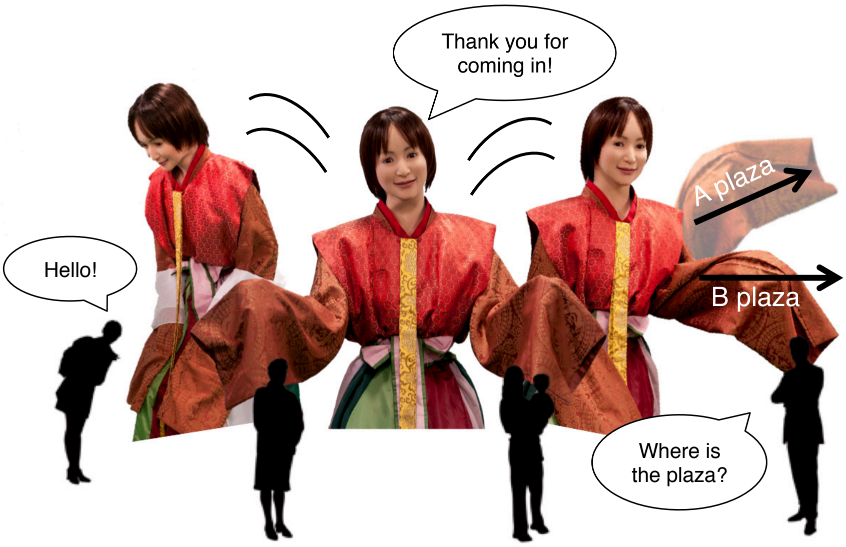


Figure 1.2: Concept of the proposed HRI system which can communicate with multiple people by interruptible communication and gesture adjustment based on a speaker or object location. To realize these functions in real-time is the purpose of this system.

experiments. These experiments contain not only quantitative evaluations such as the measurement of response/residence time of communication but also qualitative evaluations such as the analysis of human impressions of a robot.

1.3 Thesis Layout

The rest of this thesis is organized as follows.

- **Chapter 2**
  Introduce related works about methods to generate human-like motions for robots, and their applications to HRI systems.

- **Chapter 3**
  Describe the detail of the *Reconfigurable Motion Database* (RMDB). This method has three main features: motion classification, motion parameterization, and motion interruption.

- **Chapter 4**
  Evaluate the performance of the RMDB compared with some conventional methods. We conduct experiments using one-hour daily interaction motions of 15 subjects.

- **Chapter 5**
  Describe the details of a novel multi-person interaction system implemented on an android, *Actroid-SIT*. RMDB is embedded into the system for body gesture. In addition, our system can generate facial expression and gaze movement.

- **Chapter 6**
  Show the results of multi-party HRI experiments. Over 1,700 subjects in total took part in the evaluations to compare significant differences among the proposed system and conventional systems.

- **Chapter 7**
  Conclude this thesis and give the directions of future work.

Chapter 2

Related Work

To generate human-like motion for humanoids and androids, researchers first have tried to generate the motion by mathematical approaches, such as computational and probabilistic approaches. In recent years, various data-driven methods have been proposed, because computational performance is rapidly getting better and better. Signal processing methods are also applied to the motion processing methods. And the applications to HRI systems are also widely attracted attention.

2.1 Mathematical Approaches

The mathematical approaches mean that the human-like motion is generated by mathematical models such as a computational neuroscience and probabilistic method. Flash *et al.* [10] proposed a method for reaching movement of upper limb based on a minimum jerk model. Uno *et al.* [25] also proposed a method for the reaching movement based on a minimum torque-change model. Chikaraishi *et al.* [11] proposed a method for a natural idling motion (*i.e.,* a robot moves based on an attractor selection model [26]) with online human tracking. These models are suitable for the specific motions. However, it is difficult to design empirical models for all gestures which have a wide variety of motion expressions.

In probabilistic methods, a *Probabilistic Roadmap Method* (PRM) [18] is one of the motion planning methods. A lot of probabilistic methods inspired by the PRM have been proposed such as a *Rapidly-exploring Random Tree* (RRT) [27], RRT-connect [28]. Robots in real world are required to avoid collision between themselves or obstacles. The PRM inspired methods can plan a collision-free path by sampling the configuration space (C-space) of a robot. It is quite helpful when you use an industrial robot. However, they generate just a shortest path and it is difficult to generate a human-like motion.

Robots can also perform multi-modal communications such as dialog, various gestures, and facial expression. Scassellati *et al.* [29] implemented the function of tracking human gaze and pointing finger, to enable joint attention which people focus on the same object with each other. Ogawa *et al.* [30] evaluated the effectiveness of nodding while people communicate with a robot.

Generating human-like gaze movement is important to make better interaction with human. Masuko *et al.* [31] proposed a method which uses a sharing rate and a convergence of both eyes for CG agent based on the knowledge of previous researches [22, 23, 24]. It is crucial to apply the method into androids which we use. Androids are expected to be more effective for presenting multi-modal information because of their appearance unlike other robots. However, the appearance sometimes leads to human's mis-attention to the android rather than a target object, and causes debilitating memory loss against the presentation [32]. Therefore, we aimed at making sure the motions considering human attention and impression.

2.2 Data-driven Approaches

The another approach to generate human-like motions is data-driven methods. It means a database reutilization of human's motions captured by a motion capture system. We can easily capture human's motion, because optical or image based motion capturing systems are common nowadays. A *Motion Graph* [12] generates motions by smoothly connecting with similar poses between motions. A *Match Web* [17] retrieves and synthesizes similar motions. Brügmann *et al.* [33] applied a *Verbs and Adverbs* [34], which is a method for phase-based gesture parameterization that can deal with gesture interruptivity and transitions, into a CG agent. Although this approach is similar to our approach, it works only on virtual environment because of the potentiality of collision.

The *Motion Graph* and *Match Web* is suitable for our research aim. However, these methods originally focus on accurate motion connection and synthesis of walking, punching or kicking motions and classifies them based on an apparent similarity of motion using marker's positions of motion capture data. In contrast, the gestures we need are required to measure semantic similarity of motion. For example, wherever you point to, the gestures are pointing gestures even if their appearance are different between them. Therefore, we proposed a method for classifying motions in a semantic meaning. The proposed method is described motions in a frequency domain and transforms them into features. To accelerate motion retrieval, we employ the techniques of *Bag-of-words* [15], which aims at similar sentence retrieval, and *Bag-of-features* [16], which aims at similar image retrieval, for the similar gesture retrieval using the motion features.

There are some research methods to retrieve similar motions using the *Bag-of-features*. Liu *et al.* [35] proposed action classification based on image features in video clips. Raptis *et al.* [36, 37] also proposed action classification based on joint trajectories. And Ryoo [38] applied his spatio-temporal matching method [39] into the action prediction problem. Schüldt *et al.* [40] proposed a local *Support Vector Machine* [41] approach which uses the size, the frequency and the velocity of moving patterns in video clips.

2.3 Signal Processing Approaches

Hidden Markov Model (HMM) [14] is a powerful statistical tool for modeling generative sequences (*i.e.,* signals) that can be characterized by an underlying process generating an observable sequence. After development of a software *Hidden Markov Model Toolkit* [1] which is a portable toolkit for building and manipulating HMM, the HMM has found application in many areas interested in signal processing, and in particular speech processing. HMM-based speech synthesis methods have also been studied by many researchers (*e.g.,* [42, 43, 44, 45, 46]).

Motion trajectory which we use is regarded as a kind of signal. Therefore, motion synthesis methods can utilize the HMM-based synthesis methods. Inamura *et al.* [13] use the HMM to blend different motions. They also applied to a recognition and teaching of human's motion by the HMM [47]. Although this method is difficult to synthesize more than two motions at a time, to treat a motion as a kind of signal is also very informative for the construction of RMDB.

2.4 Applications to Human-Robot Interaction System

Previous research methods in HRI mainly focus on one-to-one communication. Ido *et al.* [48] developed a Question & Answer (Q & A) communication system which can react based on a human's speech and gaze tracking. Sakamoto *et al.* [21] developed a tele-communication system which can remotely control an android using facial motion capture. HRI is also required physical interaction. In application for physical HRI, Lee *et al.* [49] proposed a mimetic communication model with impedance control to acquire

[1] http://htk.eng.cam.ac.uk/

motion primitives through the imitation learning. Haddadin *et al.* [50] also proposed physical HRI which uses collision information to between a robot and a human to achieve their cooperative tasks. These research methods were effective in a limited task domain; however, the expansion of the task domain makes the preparation of all the gestures unfeasible.

Mobile service robots also have been studied by many researchers (*e.g.,* [51, 52, 53, 54]). Although the robots can approach people to be face-to-face, the human-like expression by a robot is out of their interests.

People expect to employ androids more effectively for presenting multi-modal information because their appearance resembles human compared to other robots. We believe that an android can also be used as a tele-communication medium. Previous media, such as video conference systems, have problems with effective presence; since people do not feel they are sharing physical space [55], it is hard to identify gaze [56] and so on. And it is unlikely that only one speaker can communicate with the android at one time. Matsusaka *et al.* [20, 57] proposed how the robot can participate in multi-party conversations. This system can estimate the current speaker and the next speaker in the party by gaze tracking and speech recognition, and react to improve awareness of the robot. Nakanishi *et al.* [58], and Sakamoto and Ono [59] proposed how to construct relations between agents and humans or between robots and humans using the psychology of interpersonal relations. These research methods mainly focused on recognizing the multi-party conversational situation. We are now developing a system on which RMDB is implemented to enable not only recognition, but also interaction for multiple people with human-like gestures and facial expression.

Chapter 3

Reconfigurable Motion Database

3.1 Concept

Our proposed motion planning method, a *Reconfigurable Motion Database* (RMDB) can generate flexible gestures in real-time. RMDB is based on the *Motion Graph* [12] and *Match Web* [17] which are proposed by Kovar *et al*. Originally, these methods aimed at CG animation. To apply these methods to a real robot, however, we need to additionally give the methods the ability of real-time collision avoidance. To realize that, we propose a HRI-oriented planning algorithm inspired by the RRT [27].

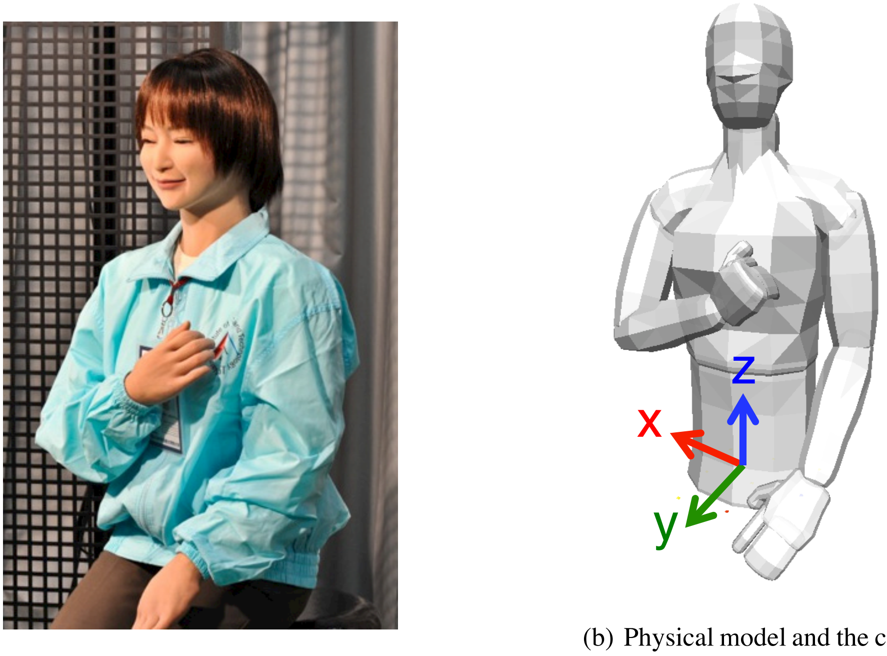
In addition, the *Motion Graph* and *Match Web* focus on accurate motion connection and synthesis based on an apparent similarity. In contrast, the gestures are required to measure semantic similarity of motion. Therefore, we proposed a novel method for classifying motions in a semantic meaning. The proposed method is described motions in a frequency domain and transforms them into features. To accelerate motion retrieval, we employ the techniques of the *Bag-of-words* [15] technique for the similar gesture retrieval using the motion features.

3.2 An Android, Actroid-SIT

In HRI, we consider human impressions affected by the appearance of a robot as well as the robot's body gestures. Recently, a very human-like robot, or an android has been developed [60]. In this research, we use one of the android, *Actroid-SIT* made by Kokoro company, Ltd. The robot has a human-like figure as shown in Figure 3.1 (a) and total of 42 degree of freedoms (DOFs) [1] as shown in Table 3.1. Because the Actroid is driven by pneudraulic actuators, damage caused by accidental collision is much less than with other motor-drive rigid robots.

For physical simulation (*e.g.,* self-collision check and kinetic simulation based on a proportional derivative (PD) control), we construct a simulation model as shown in

[1] 29 DOFs for body gesture, and 13 DOFs for facial expression



(a) Actroid-SIT

(b) Physical model and the coordinate system in the simulation world

Figure 3.1: An android, *Actroid-SIT*: (a) its figure and (b) its physical simulation model

Table 3.1: The configuration of the DOFS which the Actroid and the physical model has for each body part

body part	DOFs	
	Actroid-SIT	physical model
Face	13	0
Neck	4	2
Arms	9×2	9×2
Hands	2×2	2×2
Torso	3	2
Total	42	26

Figure 3.1 (b). Since we concentrate on a body gesture, not facial expression, we exclude DOFs corresponding the facial motion. Reduction of DOFs accelerates the collision detection.

3.3 Motion Classification

3.3.1 Bag-of-Words' Approach

Bag-of-words model is a simplifying representation used in natural language processing and information retrieval. In this model, a text such as a sentence or a document is represented as an unordered collection of words, disregarding grammar and even word order. The bag-of-words model is commonly used in methods of document classification, where the frequency of each word is used as a feature for training a classifier.

Recently, the *Bag-of-words* model has also been used for computer vision. It is called *Bag-of-features*. In this model, a image is represented as an unordered collection of local image features such as a *Scale Invariant Feature Transformation* (SIFT) [61], *Features from Accelerated Segment Test* (FAST) [62], or *Speeded-Up Robust Features* (SURF) [63].

3.3.2 Spatio-Temporal Motion Feature

When calculating a similarity of motions, it is important to not only focus on an instant pose but also express a relationship between local motion sequences. In image processing, for expressing local spatial information of an image, the image is described in frequency domain.

This idea can be applied into the motion classification. We use a wavelet as a local temporal feature of motion. The procedure of transforming from the coordinate data of a motion $\boldsymbol{p}$ to a motion feature $\boldsymbol{f}$ is described as follow. First, given a three-dimensional motion sequence $\boldsymbol{p}(t)$ which has N motion capture markers (*i.e.,* $3 \times N$ dimensional vector) and has already calibrated by a method described in Appendix A, the velocity $\dot{p}^i_{\{x,y,z\}}(t)$ for marker i is transformed to a wavelet $f^i_{\{x,y,z\}}(\omega, t)$ by a continuous wavelet transform $\psi(t)$ in Equation (3.2). We used *Morlet* wavelet [64] as the $\psi(t)$.

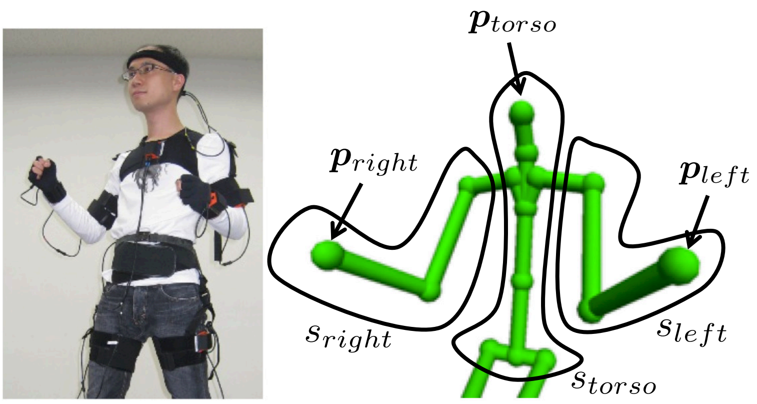


Figure 3.2: The skeleton model captured by the sensor suit *MVN*, and its three subsets of retro-reflective markers of a human's torso, left and right arms

$$\boldsymbol{p}(t)^{\mathrm{T}} = \left(\boldsymbol{p}_1(t)^{\mathrm{T}} \quad \ldots \quad \boldsymbol{p}_N(t)^{\mathrm{T}}\right), \tag{3.1}$$

$$f^i_{\{x,y,z\}}(\omega, t) = \frac{1}{\sqrt{\omega}} \sum_{t'} \dot{p}^i_{\{x,y,z\}}(t')\psi\left(\frac{t'-t}{\omega}\right), \tag{3.2}$$

where ω is a frequency scale of the wavelet. Since we focus on body gestures, we only deal with markers of the upper-body. We divide markers on upper body into three subsets: torso (s_{torso}), left arm (s_{left}), and right arm (s_{right}) as shown in Figure 3.2. Then, the frequency spectrum $s_{torso}(\omega, t)$ is calculated by Equation (3.3). After calculating Equation (3.3) within $0 \le \omega \le \Omega$ where Ω is a maximum value of the frequency scale, the movement (*i.e.,* temporal) feature $\boldsymbol{m}(t)$ is obtained as shown in Equation (3.5). $\boldsymbol{m}_{left}(t)$ and $\boldsymbol{m}_{right}(t)$ can be calculated in similar manner from the sets s_{left} and s_{right}, respectively.

$$s_{torso}(\omega, t) = \left| \sqrt{\sum_{i \in S_{torso}} f^i_x(\omega, t)^2 + f^i_y(\omega, t)^2 + f^i_z(\omega, t)^2} \right|, \tag{3.3}$$

$$\boldsymbol{m}_{torso}(t)^{\mathrm{T}} = \left(s_{torso}(0, t) \quad \ldots \quad s_{torso}(\Omega, t)\right), \tag{3.4}$$

$$\boldsymbol{m}(t)^{\mathrm{T}} = \left(\boldsymbol{m}_{torso}(t)^{\mathrm{T}} \quad \boldsymbol{m}_{left}(t)^{\mathrm{T}} \quad \boldsymbol{m}_{right}(t)^{\mathrm{T}}\right). \tag{3.5}$$

We use the appearance (*i.e.,* spatial) feature $\boldsymbol{a}(t)$ which is a coordinate data of the end-effector's markers shown in Figure 3.2 $\boldsymbol{p}_{\{torso,left,right\}}$.

$$\boldsymbol{a}(t)^{\mathrm{T}} = \left(\boldsymbol{p}_{torso}(t)^{\mathrm{T}} \quad \boldsymbol{p}_{left}(t)^{\mathrm{T}} \quad \boldsymbol{p}_{right}(t)^{\mathrm{T}}\right). \tag{3.6}$$

Finally, we obtain the spatio-temporal motion feature $\boldsymbol{f}(t)$ which includes the movement feature $\boldsymbol{m}(t)$ and the appearance feature $\boldsymbol{a}(t)$ by Equation (3.7).

$$\boldsymbol{f}(t)^{\mathrm{T}} = \left(\boldsymbol{m}(t)^{\mathrm{T}} \quad \boldsymbol{a}(t)^{\mathrm{T}}\right). \tag{3.7}$$

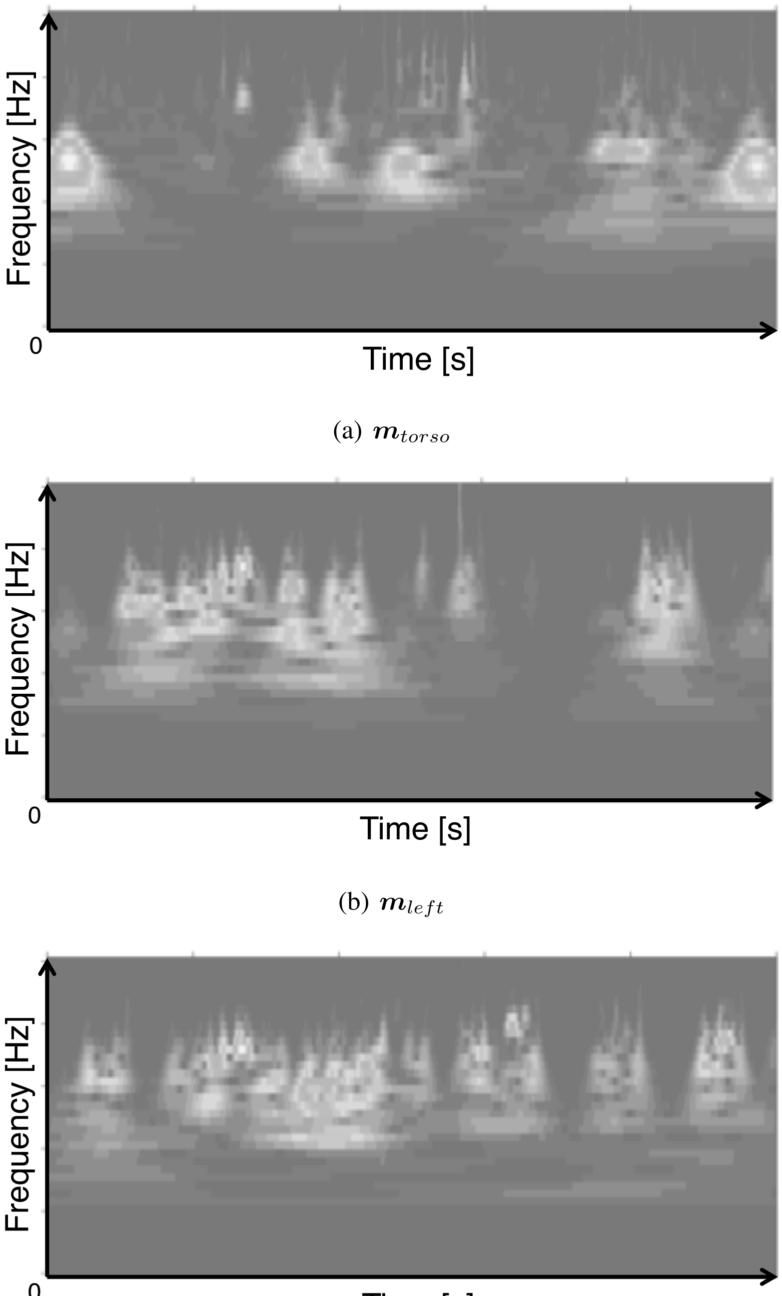
The motion feature $\boldsymbol{f}(t)$ is independent of the number of markers N. That is, the feature is independent of the skeleton model and we can unify the variety of motion data captured by different motion capturing systems. We have three subsets and the frequency resolution is 40. $\boldsymbol{f}(t)$ is a $3\times40+3\times3 = 129$ dimensional vector. We use a sensor suit *MVN* made by Xsens Technologies, Inc. shown in Figure 3.2 for measuring human's motions.

Figure 3.3 shows the visualization of the movement features $\boldsymbol{m}_{\{torso,left,right\}}$ of one-minute gesture motion of a subject. The horizontal axis indicates the time and the vertical axis indicates the frequency. The frequency spectrum is higher where the color is brighter. A slow movement appears in a low frequency area and a quick movement appears in a high frequency area. Both of them provides important information to describe how a gesture can be expressed.

3.3.3 Bag-of-Motion-Features

For motion classification and retrieval, we employed an idea of image processing technique: the *Bag-of-features*. First, the motion feature $\boldsymbol{f}$ is discretized (*i.e.,* cluster) by K-means method. At this time, the six factors $\boldsymbol{m}_{\{torso,left,right\}}$ and $\boldsymbol{a}_{\{torso,left,right\}}$ are clustered independently.

Unlike natural language classification or image classification, motion classification additionally considers the order of the motion. Therefore, we applied the idea of *Spatial Pyramid Kernel* [65] which uses the hierarchical histograms of evenly divided lattices of an image. Unlike the method [65], we construct hierarchical structure along the time axis. The motion feature $\boldsymbol{f}$ is evenly divided several times as shown in Figure 3.4. The histogram intersection between level l's motions $\mathscr{A}, \mathscr{B}$ is calculated by Equation (3.8) based on the histogram intersection function [66].



(a) $\boldsymbol{m}_{torso}$

(b) $\boldsymbol{m}_{left}$

(c) $\boldsymbol{m}_{right}$

Figure 3.3: The visualization of (a) $\boldsymbol{m}_{torso}$, (b) $\boldsymbol{m}_{left}$, and (c) $\boldsymbol{m}_{right}$ of a one-minute motion. The spectrum of the frequency is higher where the color is brighter.

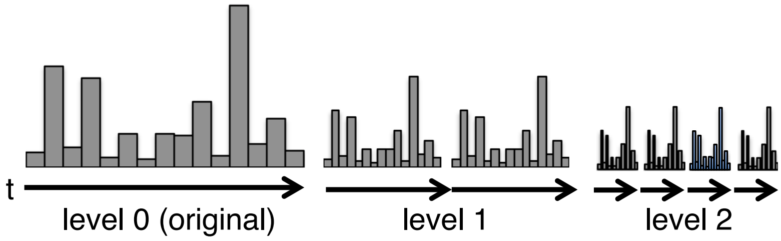


Figure 3.4: The example of a three-level temporal pyramid histogram

$$\mathcal{I}_l(\mathscr{A},\mathscr{B}) = \frac{\sum_{i=1}^{K} \min\left(\frac{H^l_{\mathscr{A}}(i)}{w_{\mathscr{A}}}, \frac{H^l_{\mathscr{B}}(i)}{w_{\mathscr{B}}}\right)}{\sum_{i=1}^{K} \frac{H^l_{\mathscr{A}}(i)}{w_{\mathscr{A}}}}, \tag{3.8}$$

where $H^l_{\mathscr{A}}(i)$ indicates i-th bin of the level l's histogram of a motion $\mathscr{A}$ and K indicates the number of bins of the histogram. $w_{\mathscr{A}}, w_{\mathscr{B}}$ are the number of frames of each motion. By calculating weighted average of all $\mathcal{I}_l(\mathscr{A},\mathscr{B})$, we finally obtained the histogram intersection $\mathcal{I}(\mathscr{A},\mathscr{B})$ by Equation (3.9).

$$\mathcal{I}(\mathscr{A},\mathscr{B}) = \frac{1}{2^L}\mathcal{I}_0(\mathscr{A},\mathscr{B}) + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}}\mathcal{I}_l(\mathscr{A},\mathscr{B}). \tag{3.9}$$

We name this method the *Bag-of-motion-features*.

3.4 Motion Parameterization

3.4.1 Match Web's Approach

Match Web is a method for motion synthesis. Given a human-like motion sequence, which is easily obtained from a motion capture system, the *Match Web* can extract similar sequence duration from all sequences. The details of the algorithms are as follows.

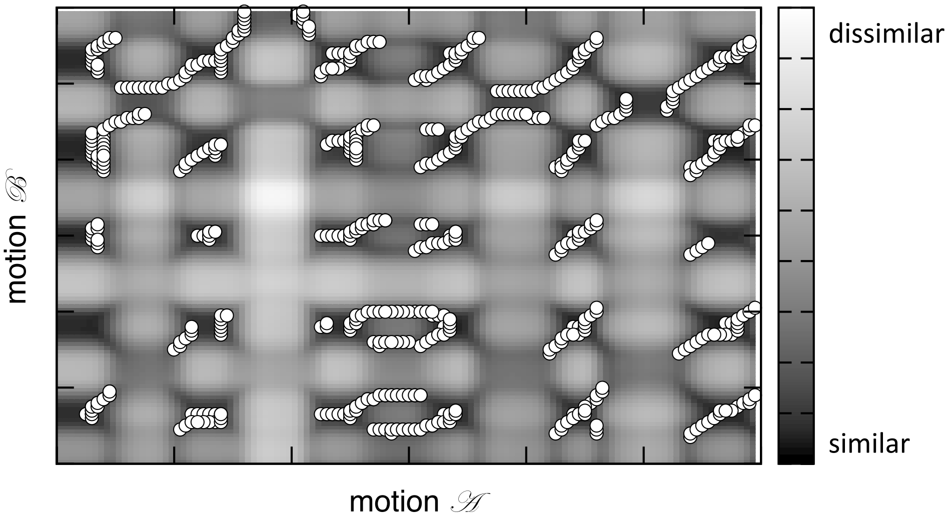


Figure 3.5: Calculation of similar regions between two motion sequences. The similarity is higher where it is black and lower where it is white, and white circles are similar region.

1. Given two motion sequences $\mathscr{A}$ and $\mathscr{B}$, calculate the similarity matrix between poses $\mathscr{A}_i$ and $\mathscr{B}_j$ (Figure 3.5 cells), in sequences $\mathscr{A}$ and $\mathscr{B}$.
2. If the similarity $D(\mathscr{A}_i, \mathscr{B}_j)$ is a local minimum, attempt to connect other local minima from the bottom left to the top right in the similarity matrix along less dissimilar path using a dynamic programming (DP) matching. This path indicates similar region (Figure 3.5 dotted line).
3. The user assigns a meaningful motion region (*i.e.* reference motion) from a sequence, then similar regions (*i.e.* similar motions) are extracted by searching all similar regions.

This procedure is repeated for all pairs in case of multiple sequences. Then, given an arbitrary parameter for each motion group, target motion is synthesized using the similar motions based on a target parameter.

3.4.2 Coarse-to-Fine Motion Classification

The original *Match Web* focuses on accurate motion synthesis based on an apparent similarity and is also time-consuming to calculate the DP matching. To solve these problems, we use the *Bag-of-motion-features* which can rapidly classify based on the semantic similarity.

In Section 3.3.3, the motion feature $\boldsymbol{f}$ of a motion needs to be divided into primitives by constant time windows to make histograms. This can cause a classified motion to contain movements unrelated to a target gesture during starting time or ending time of the time window. There are some solutions to this problem. A simple solution is to use a finer time window. However, it would need longer computational time. Another solution is to apply the *Match Web* technique to the search region limited by the *Bag-of-motion-features*.

In this research, we use the latter approach. After performing motion retrieval by the *Bag-of-motion-features*, the retrieved motion is matched with the DP matching in detail. Because the detail matching is only for the retrieved motions, we expect the total of the computational time to be still short.

3.4.3 Parametric Motion Synthesis

Given an arbitrary parameter for the gestures, target gestures are synthesized using similar motions based on the target parameter $\tilde{\mathbf{p}}$ as shown in Table 3.2. When calculating the weighted sum of each i-th similar motion, the weight coefficient w_i for $\tilde{\mathbf{p}}_\mathbf{i}$, which is the gesture parameter of the i-th motion, is given by Equation (3.10), where c is a constant related to the variance of the parameter $\tilde{\mathbf{p}}$. We use Welsch's weight function [67] as w_i.

$$w_i = \frac{\exp\left(-\frac{D(\tilde{\mathbf{p}},\mathbf{p_i})^2}{c}\right)}{\sum_{j=1}^{k}\exp\left(-\frac{D(\tilde{\mathbf{p}},\mathbf{p_j})^2}{c}\right)}. \tag{3.10}$$

Original *Match Web* calculates the weighted coefficient by computing a bounding box of the parameter space and pre-sampling parameters in the space to precisely synthesize motions. This pre-sampling method is time-consuming. And when the number of similar motions is getting more, we do not have to compute this pre-sampling. Therefore, we use the concise Equation (3.10).

In addition, unlike CG research, the Actroid is driven by the pneudraulic actuators, and suddenly changing acceleration of the actuator vibrates motion in the corresponding part. Therefore, similar motions are aligned using *Dynamic Time Warping* [68] to minimize poses' differences. In addition, real-time self-collision avoidance is crucial issue. To solve the issue, we propose an online motion planning method in Section 3.5.

In this thesis, the RMDB has 18 gestures, each of which has more than 10 similar motions in the motion sequences which is described in Section 4.1.1. Table 3.2 shows the 18 gestures and their parameters. The terms x, y, z indicate the 3D position of target, ϕ, θ indicate yaw (horizontal) and pitch (vertical) angles of the robot's direction, and l indicates the breadth between both hands. Gestures #7, #9, #11, #14, #17, #18 which use only one hand, are archived by either the left hand or the right hand. RMDB can select an appropriate hand based on the target parameter $\tilde{\mathbf{p}}$. And gesture #1 performs to track speaker's direction by combining gaze described in Appendix B.

Figure 3.6 shows examples of a pointing gesture #18 POINT where the target parameter (x, y, z) is shifted from the right side to the left horizontally. This figure shows precise motion synthesis is achieved while keeping the twist motions of the wrist which simulate actual human motion.

3.5 Motion Interruption

3.5.1 Motion Graph's Approach

In CG field, many approaches were proposed for entire body animation using motion capture data. One of them, *Motion graph* [12] can generate new motion sequence by concatenating motions at different timing. The details of the algorithms are as follows.

1. Given the two motion sequences $\mathscr{A}$ and $\mathscr{B}$, the algorithm first calculates the similarity between poses $\mathscr{A}_i$ and $\mathscr{B}_j$, in sequences $\mathscr{A}$ and $\mathscr{B}$ as shown in Figure 3.7.

2. If the similarity $D(\mathscr{A}_i, \mathscr{B}_j)$ is locally minimized (the white circles in Figure 3.7), the algorithm attempts to connect two motions around these two poses by Equation 3.11.

Table 3.2: 18 gestures and their parameters. The terms x, y, z in parameter $\tilde{\mathbf{p}}$ indicate the 3D position of target, ϕ, θ indicate yaw (horizontal) and pitch (vertical) angles of the robot's direction, and l indicates the breadth between both hands.

gesture		movement	parameter $\tilde{\mathbf{p}}$
#1	GAZE	Gaze a speaker	(ϕ, θ)
#2	YES (NOD)	Nod several times	
#3	NO	Nod no	
#4	BOW	Bow	
#5	SHAKE_NECK	Shake neck several times	
#6	THINK	Fold both arms	
#7	FLEX_MUSCLE	Flex arm muscles	
#8	POINT_AT_MYSELF	Point at myself	
#9	BYEBYE	Wave hand several times	
#10	BYEBYE_BOTH	Wave both hand several times	
#11	SWING	Swing hand	
#12	SWING_BOTH	Swing both hands	
#13	LOOK_AT_CLOCK	Look at wrist clock	
#14	SHAKE	Shake arm quickly	(ϕ, θ, l)
#15	SHAKE_BOTH	Shake both arms quickly	
#16	SPREAD_BOTH	Spread both arms widely	
#17	PUNCH	Punch something	(x, y, z)
#18	POINT	Point at something	(x, y, z, ϕ, θ)

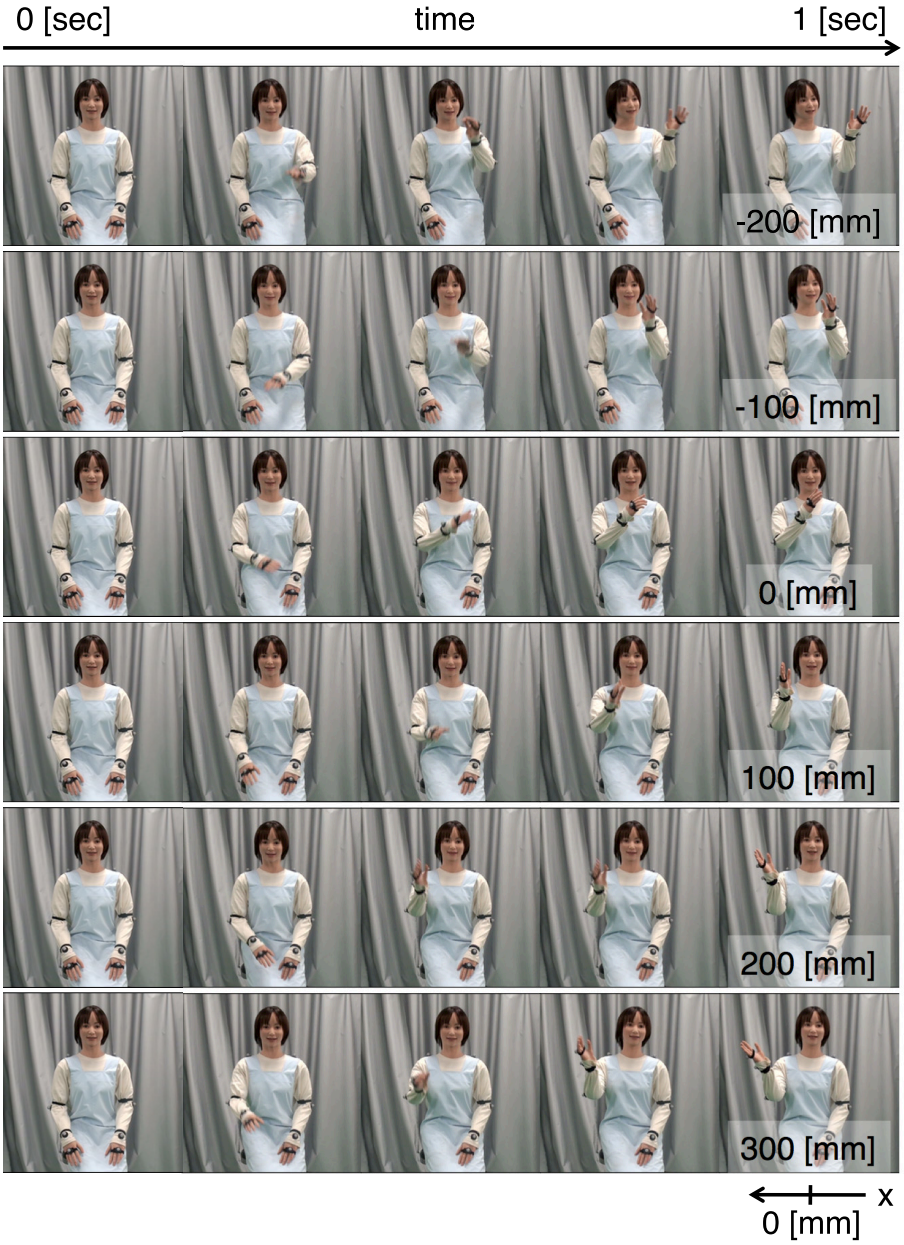


Figure 3.6: Synthetic pointing motions where pointing horizontal location x is ranged $-200\ [mm] \leq x \leq 300\ [mm]$

$$\mathscr{C}_k = \alpha(k)\mathscr{A}_{i+k} + \Big(1 - \alpha(k)\Big)\mathscr{B}_{j-m+k}, \tag{3.11}$$

$$\alpha(k) = 2\Big(\frac{k+1}{m}\Big)^3 - 3\Big(\frac{k+1}{m}\Big)^2 + 1, \tag{3.12}$$

where i, j, and k are frame index, and $\mathscr{C}$ contains, and m is user-defined connection length. The similarity function $D(\mathscr{A}_i, \mathscr{B}_j)$ is calculated by appropriately weighting similarity of body part [12]. However, the motion database which we used has a number of different gestures and it is difficult to set the appropriate weights for each one of them. That is why, we used uniform weights for all gestures in the following experiments.

Figure 3.8 shows the example of a *Motion Graph* which contains 17 receptionist's gestures for the Actroid. The nodes indicates local minima of motions and edges indicates motions between node to node.

3.5.2 Probabilistic Roadmap Method's Approach

Probabilistic Roadmap Method (PRM) [18] is a motion planning method, which samples the configuration space (C-space) of a robot. One of the sampling algorithms, *Rapidly-exploring Random Tree* (RRT) [27] is designed for efficiently searching nonconvex high-dimensional spaces. RRT is constructed incrementally in a way that quickly reduces the expected distance of a randomly-chosen point to the tree. RRT is particularly suited for path planning problems that involve obstacles and differential constraints (nonholonomic or kinodynamic). RRT can be considered as a technique for generating open-loop trajectories for nonlinear systems with state constraints. An RRT can be intuitively considered as a Monte-Carlo way of biasing search into largest Voronoi regions. Some variations can be considered as stochastic fractals. Usually, an RRT alone is insufficient to solve a planning problem. Thus, it can be considered as a component that can be incorporated into the development of a variety of different planning algorithms.

Figure 3.9 shows an example of the path planning by the RRT with lattice-shaped obstacles on two dimensional C-space. The root of the tree placed left below.

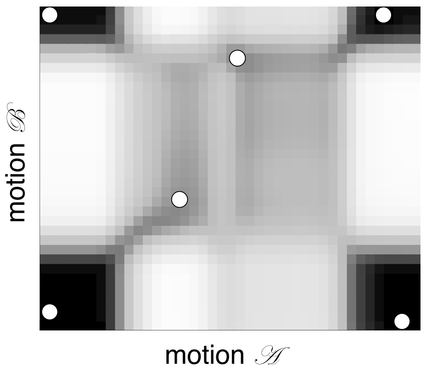


Figure 3.7: The distance of poses between motion sequence 𝒜 and ℬ. The darker area is closer pose and white circles are local minima.

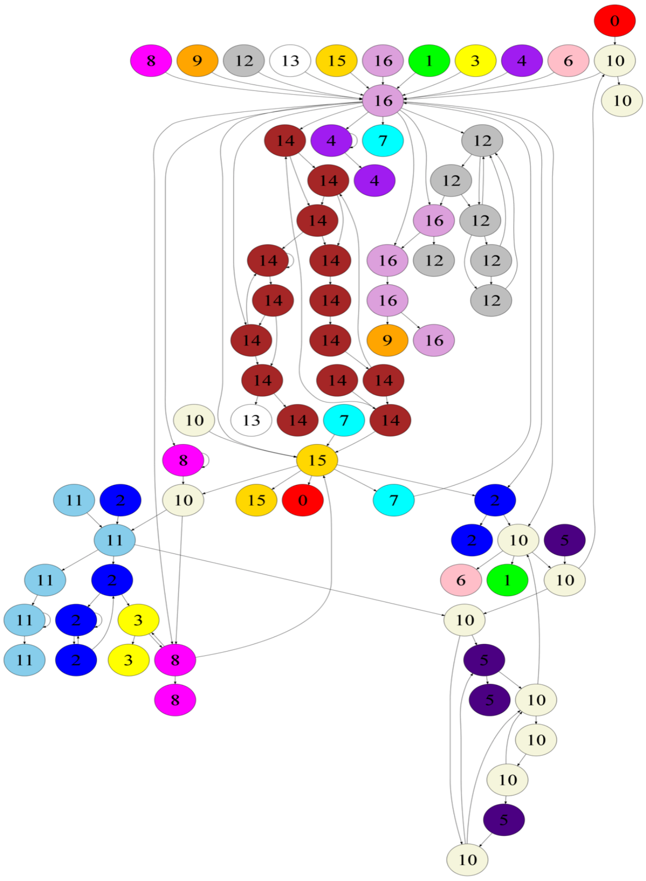


Figure 3.8: A *Motion Graph* of 17 receptionist's gesture sequences for the Actroid. The node indicates a transitional timing of a motion and the edge indicates a motion between node to node.

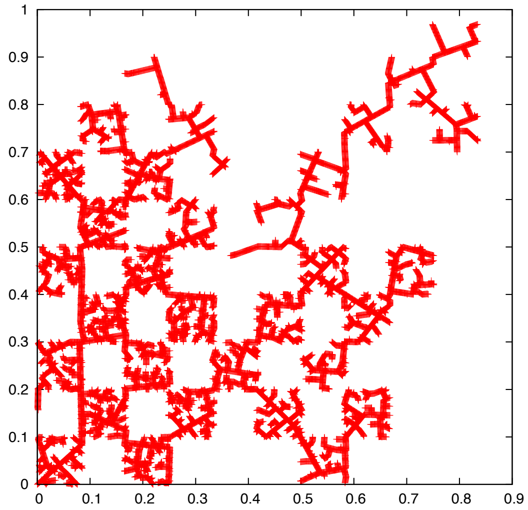


Figure 3.9: Two dimensional C-space expansion with lattice-shaped obstacles by the *Rapidly-exploring Random Tree*. The root of the tree placed left below.

3.5.3 Motion Feature

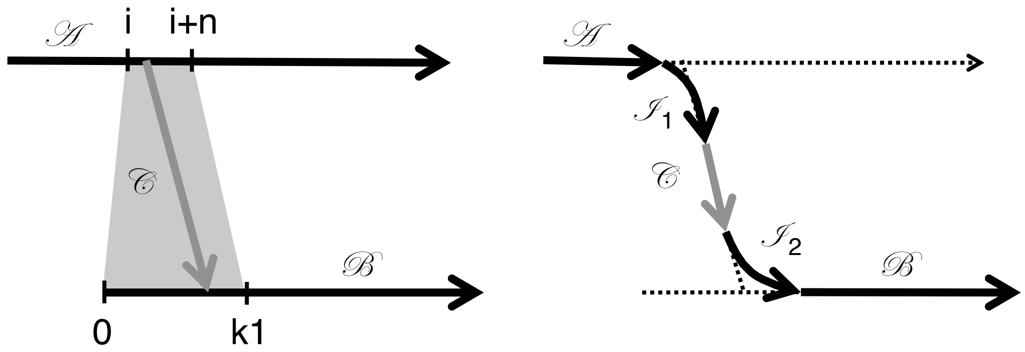
First, we define motion features appropriate for HRI; as far as we know, there is no common expression of motion features. Hereby, RMDB can reconfigure motion sequences while keeping following two features of original human's motion in database.

Key-Pose Information

Key-poses are defined as important instant pose in gesture, for example, the pose to express object position in pointing or reaching gesture. Thus, lack of key-pose information causes to become unnatural and/or meaningless motions. Nakazawa *et al.* [69] proposed the synthesized human-like dancing motions using key-pose information from captured human motions.

Velocity Information

Flash *et al.* [10] proposed a minimum-jerk model as inherent feature of physiologically-based trajectory. That is, velocity gradient is important factor for description physiologically-based motion features.



(a) Limitation of transitional area

(b) Generation of connecting motion

Figure 3.10: Planning of smooth transition from $\mathscr{A}_i$, which indicates a pose when interrupted, to $\mathscr{B}_{k1}$, which indicates first key-pose in motion $\mathscr{B}$

3.5.4 Interruptible Motion Planning

Whenever a user interrupts current robot's reaction, it is necessary to suddenly terminate the reaction and then generate and preform appropriate reaction. Even when the Euclidean distance $D(\mathscr{A}, \mathscr{B})$ is larger than the threshold, two motions should be connect while avoiding self-collision. Therefore we solve this problem by PRM. As shown in Figure 3.10, when the Euclidean distance $D(\mathscr{A}_i, \mathscr{B}_j)$ is less than the threshold, we can apply original *Motion Graph*. In contrast, when its value is larger than the threshold, we let $\mathscr{A}_i$ and $\mathscr{B}_j$ connect by motion $\mathscr{C}$ generated by PRM instead of original *Motion Graph* approach. Motions $\mathscr{A}$ to $\mathscr{C}$ and motions $\mathscr{C}$ to $\mathscr{B}$ are calculated by original *Motion Graph* method. $\mathscr{B}_{k1})$ indicates a first key-pose in the motion $\mathscr{B}$ to smoothly translate.

Multi-Subtree Rapidly-Exploring Random Trees

In HRI, trajectories without self-collision have to be planned in real-time. However, PRM is generally computationally expensive with over few dozens of dimensions of C-space.

We therefore propose a multi-subtree RRT planner. This method is inspired by the RRT and RRT-connect [28], and it can conduct with multi-core CPU processing efficiently in place of the algorithm for distributed processing [70]. The main algorithm

Algorithm 1 MULTITHREADED_RRT(q_{init}, q_{goal})

```
BUILD_SUBTREES($q_{init}, q_{goal}$)
while $\mathcal{C}$.size() $\leq$ MAX_CONFIGS do
  EXTEND_AND_INTEGRATE($\mathcal{S}, \mathcal{C}$)
  if SEARCH_AND_CHECK($\mathcal{S}, \mathcal{C}$) then
    return SMOOTH(DIJKSTRA_PATH($q_{init}, q_{goal}$))
  end if
end while
```

Algorithm 2 BUILD_SUBTREES(q_{init}, q_{goal})

```
$\mathcal{S}$.add(BUILD_TREE($q_{init}$))
$\mathcal{S}$.add(BUILD_TREE($q_{goal}$))
for $i = 3$ to NUM_THREADS do
  $\mathcal{S}$.add(BUILD_TREE(RANDOM_CONFIG()))
end for
```

Algorithm 3 EXTEND_AND_INTEGRATE($\mathcal{S}, \mathcal{C}$)

```
for $i = 1$ to NUM_THREADS do
  $\mathcal{A}[i]$.clear()
  EXTEND($\mathcal{T}[i], \mathcal{S}[i], \mathcal{A}[i]$)
end for
for $i = 1$ to NUM_THREADS do
  $\mathcal{C}$.add($\mathcal{A}[i]$)
end for
```

Algorithm 4 SEARCH_AND_CHECK($\mathcal{S}, \mathcal{C}$)

```
for $i = 1$ to NUM_THREADS do
  $\{q_1, q_2\}$ =NEAREST_PAIR($\mathcal{T}[i], \mathcal{A}[i], \mathcal{C}$)
  if DISTANCE($q_1, q_2$) $\leq$ THRESHOLD_DIST then
    $\mathcal{P}$.add($q_1, q_2$)
  end if
end for
return IS_CONNECTED($\mathcal{S}[1], \mathcal{S}[2]$)
```

Figure 3.11: Multi-subtree RRT algorithms which generate the path from initial pose q_{init} to goal pose q_{goal}. The Algorithm 1 is a main method and the Algorithm 2, 3, 4 are its sub-methods.

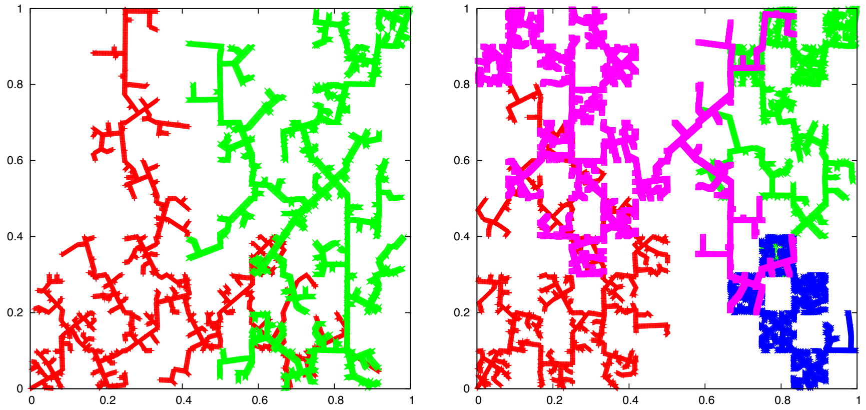
and its sub-algorithms are shown in Figure 3.11. The global variables $\mathcal{T}$ and $\mathcal{S}$ indicate a set of threads and subtrees, respectively ($\|\mathcal{T}\| = \|\mathcal{S}\| =$ NUM_THREADS). The space $\mathcal{C}$ is a whole C-space which contains all subtrees. Subtrees extends independently in each thread $\mathcal{T}[i]$. The set $\mathcal{A}$ represents the added nodes in one extension. By connectivity between $\mathcal{A}$ and each subtree, the algorithm obtains connectivity among subtrees.

Algorithm 1 shows the abstract of multi-subtree RRT. It plan the trajectory between initial pose q_{init} and goal pose q_{goal} by calculating Algorithm 3 and 4 iteratively, where the function DIJKSTRA_PATH searches the shortest path through subtrees, and finally the function SMOOTH smoothens the connected path. Algorithm 2 makes subtrees whose routes are q_{init}, q_{goal}, and what RANDOM_CONFIG generates. In Algorithm 3, each subtree extends and integrates the appended nodes in $\mathcal{C}$ concurrently. Algorithm 4 searches nearest pair between $\mathcal{A}[i]$ and $\mathcal{T}[i]$ with NEAREST_PAIR, then adds the pair whose distance is less than THRESHOLD_DIST, to the set $\mathcal{P}$ in synchronization. The detail of other functions RANDOM_CONFIG, and EXTEND, SMOOTH are described in [27, 28].

Figure 3.12 shows the results of path planning by the multi-subtree RRT with lattice-shaped obstacles on two dimensional C-space. Figure 3.12 (a) has one tree which means RRT, and Figure 3.12 (b) has two subtrees which mean RRT-connect. Figure 3.12 (c) extends more widely in the space than Figure 3.12 (a) and Figure 3.12 (b).

Exclusion of Redundant Movement in Search

In addition to the multi-subtree RRT, we assume that a motion in interaction tends to exclude redundant movement. Thus, its travel distance should be minimized. Considering this fact, we propose to limit the search area which results in accelerating the planning (under 100 [ms] ordinarily). Figure 3.13 shows abstract of our algorithm. First, a straight-line interpolation trajectory from initial pose to goal pose is given as a reference, and each range is limited to nearby areas of the reference (Figure 3.13-1). If there is collision on the way, the trajectory is re-planned in the limited area using multi-subtree RRT (Figure 3.13-2, 3.13-3). Finally, the planned trajectory is smoothed (Figure 3.13-4).



(a) Two subtrees (b) Four subtrees

Figure 3.12: C-space subtree expansions by (a) two subtrees and (b) four subtrees. The one tree example are shown in Figure 3.9.

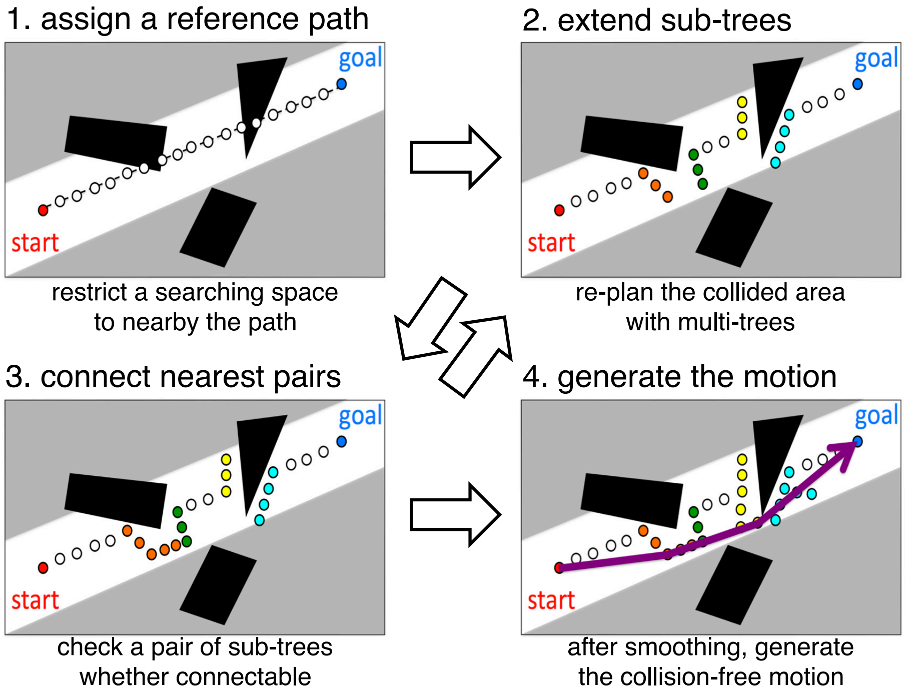


Figure 3.13: HRI-oriented planning algorithm using multi-subtree RRT (white area: collision-free, black area: obstacle, gray area: un-calculated)

Chapter 4

Database Construction and Performance Evaluation

First, we construct a motion database which contains a lot of captured human motions by the motion classification. Then, we evaluate the effectiveness of the motion parameterization and motion interruption using the database.

4.1 Motion Classification

We conduct experiments using total of one-hour motions by 15 subjects to evaluate the classification performance of the proposed method with three comparative methods including the conventional method *Match Web*. The computational time of the motion database construction is also measured.

4.1.1 Database Construction

Motion Measurements

We measured daily interaction motions of 15 subjects who are women and men in their 20s. 30-minutes motions were used for the learning data set to cluster the *Bag-of-motion-features*, another 30-minute motions were for test data. The sensor suit *MVN* was used for measurement of motions at 30 [Hz].

Clustering of Motion Features

First, we decide the numbers of clusters K. By conducting experiments to measure classification performance for all numbers of clusters of K_m for $\boldsymbol{m}_{\{torso,left,right\}}$ and K_a for $\boldsymbol{a}_{\{torso,left,right\}}$, we set the numbers of clusters to four for each $\boldsymbol{m}_{\{torso,left,right\}}$ and six for each $\boldsymbol{a}_{\{torso,left,right\}}$ (*i.e.*, $K = 3K_m{+}3K_a = 3{\times}4{+}3{\times}6 = 30$). Figure 4.1 shows the result of a *Receiver Operating Characteristic* (ROC) curve in case of the number of $\boldsymbol{m}_{\{torso,left,right\}}$ clusters to four. In ROC curve, higher true positive rate

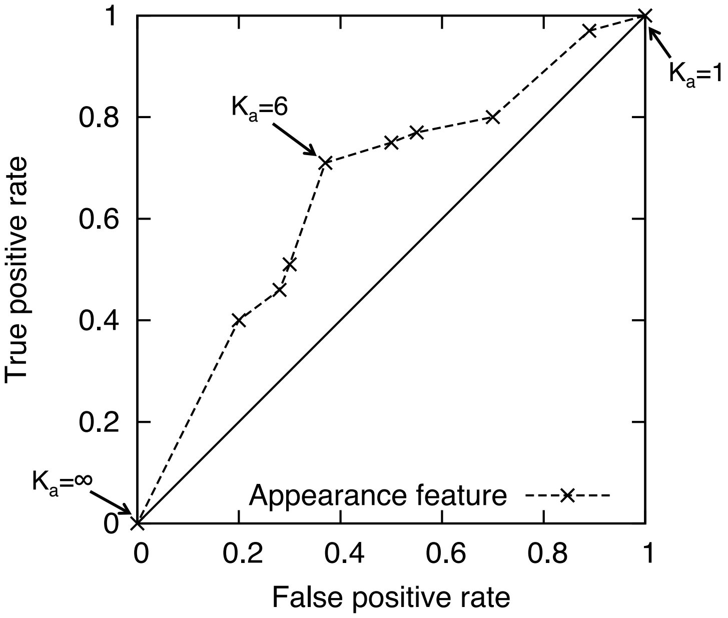


Figure 4.1: The ROC curve in case of the number of $\boldsymbol{m}_{\{torso,left,right\}}$ clusters K_m to four. The number of $\boldsymbol{a}_{\{torso,left,right\}}$ clusters K_a to six is the best condition of the true/false positive rates.

and lower false positive rate are better conditions. We confirmed that these numbers are the best condition of the true/false positive rates for our motion database.

Figure 4.2 (a), (b) show the frequencies of each cluster of the 30-minutes motion database with respect to the movement features (a) $\boldsymbol{m}_{\{torso,left,right\}}$ and the appearance features (b) $\boldsymbol{a}_{\{torso,left,right\}}$. Figure 4.2 (a) is in almost ascending order according to the frequency. That is, the leftmost and largest cluster means almost static pose. Because subjects often behave idling or waiting with forearm resting on their lap, the size of the leftmost cluster is so large.

This is the same reason why the leftmost size of the cluster in Figure 4.2 (b) is much larger than the other clusters. The other clusters indicate that the end-effector locates somewhere in Cartesian coordinates.

Motion Division with Time Window

Before constructing the *Bag-of-motion-features*, the motion feature $\boldsymbol{f}$ needs to be divided into primitives by a time window of length w, as mentioned in Equation (3.8). Gestures have a variety of time lengths. For example, nodding gesture is shorter and bye-bye gesture is longer than the other gestures. Therefore, we used the time window's length $w = 1, 2, 3, 4, 5$ [s] and shifted the time window to 0.2 [s] intervals.

Calculation of Features' Histograms

Next, we construct histograms of the *Bag-of-motion-features* using aforementioned motion feature. Figure 4.3 (a), (b) show two examples of the histograms. Figure 4.3 (a) shows a right-hand's pointing gesture #18 POINT and Figure 4.3 (b) shows a bowing gesture #4 BOW. The horizontal axis indicates a bin for each $\boldsymbol{m}_{\{torso,left,right\}}$, $\boldsymbol{a}_{\{torso,left,right\}}$ and the vertical axis indicates the frequency of each histogram's bin.

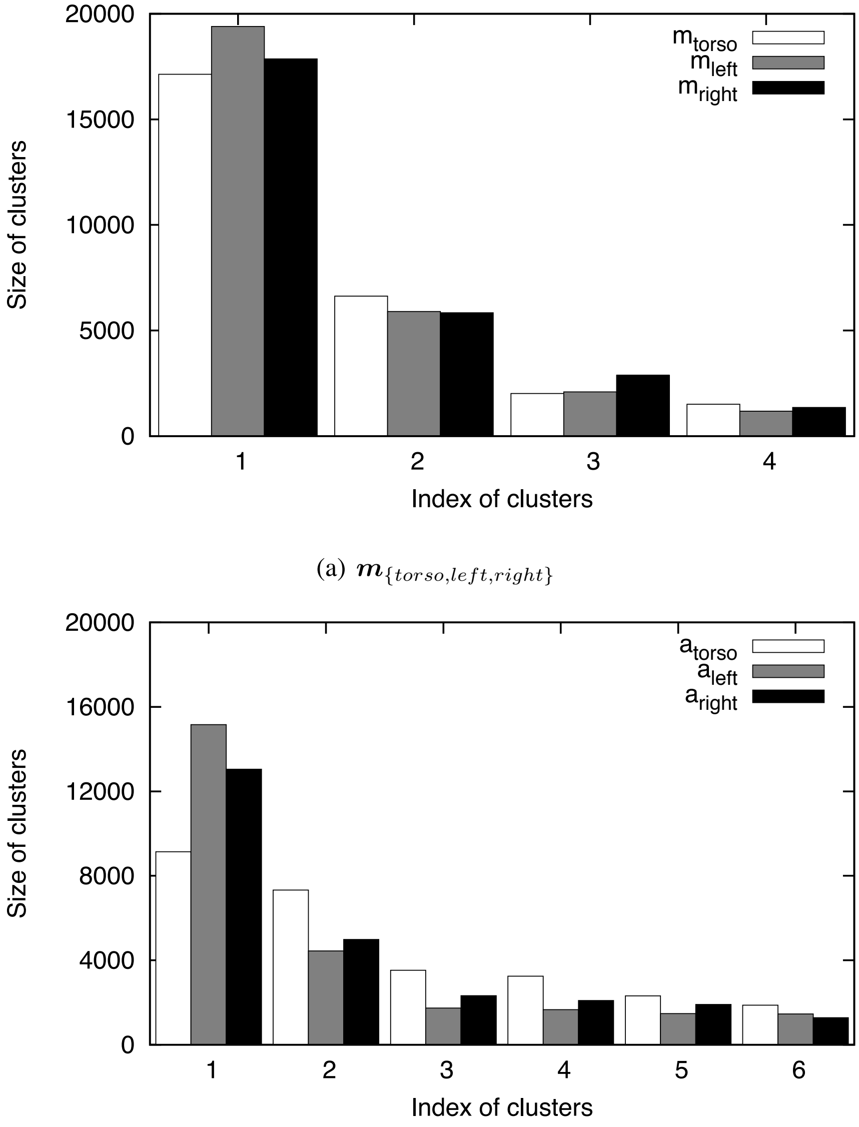
Since the histogram of $\boldsymbol{m}_{right}$ in Figure 4.3 (a) has only two activated bins, it means that the right hand moved at a constant frequency and stopped at a target pointing location. And since the histogram of $\boldsymbol{a}_{right}$ Figure 4.3 (a) has activated four bins, it means that the right hand moved a lot in Cartesian coordinates. This also applied to the histograms of $\boldsymbol{m}_{torso}$ and $\boldsymbol{a}_{torso}$ in Figure 4.3 (b), that have a similar pattern to the case of $\boldsymbol{m}_{right}$ and $\boldsymbol{a}_{right}$ in Figure 4.3 (a).

4.1.2 Comparative Methods

For evaluation of the performance of the motion classification, we compared the numbers of similar motion classification with several comparative methods. One of the comparative methods is the *Match Web* mentioned in Section 3.4.1.

According to the combination of the feature calculation method and the motion matching method between the proposed method and the *Match Web*, we have four comparative experiments as shown in Table 4.1. These methods indicate (a) our proposed method, the *Bag-of-motion-features*, (b) the *Match Web*, (c) the *Match Web* using the proposed feature vector $\boldsymbol{f}$ instead of the marker position itself $\boldsymbol{p}$ for the similarity function, and (d) the *Bag-of-features* using $\boldsymbol{p}$ instead of $\boldsymbol{f}$ for the features. The method (d) is the almost same as a method proposed by Raptis *et al.* [36].

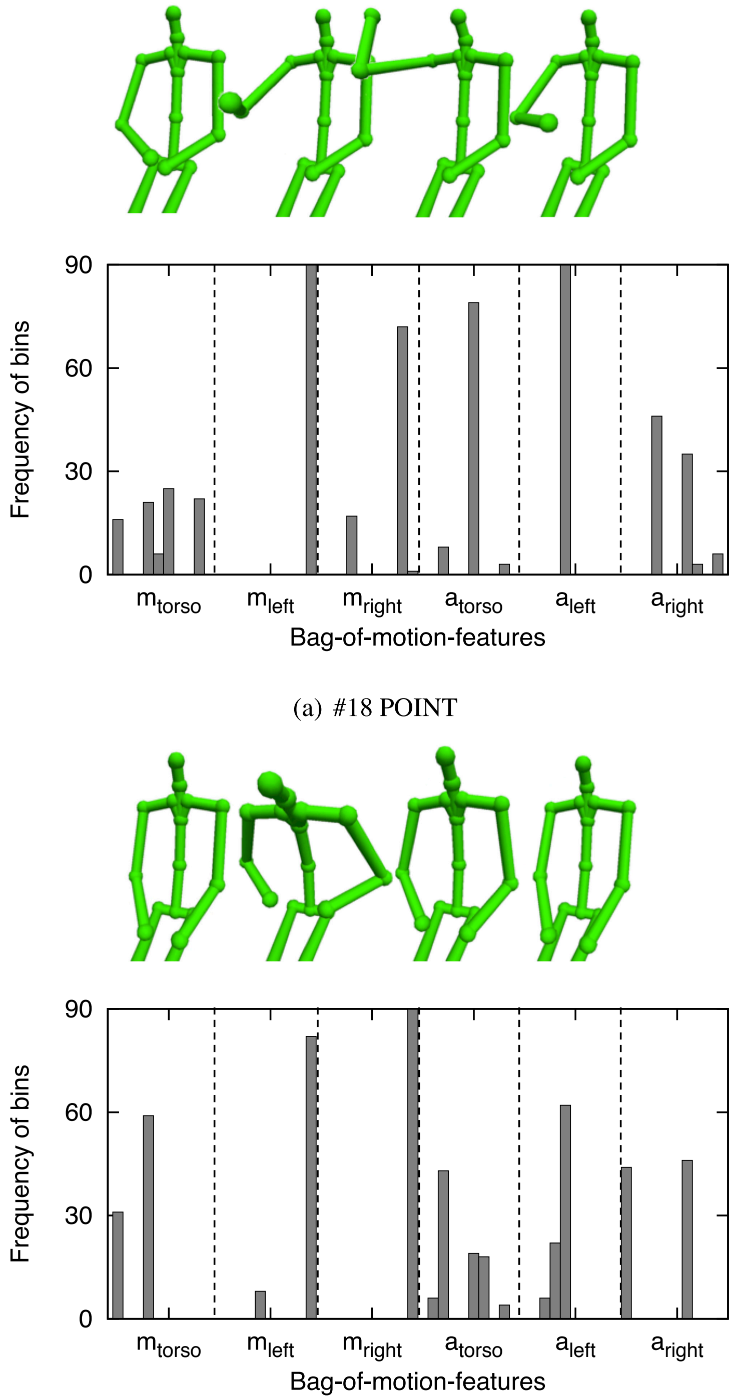
The number of the k-means clusters for (d) is $K = 60$ which is equivalent to



(a) $\boldsymbol{m}_{\{torso,left,right\}}$

(b) $\boldsymbol{a}_{\{torso,left,right\}}$

Figure 4.2: The size of each cluster with respect to the features (a) $\boldsymbol{m}$ and (b) $\boldsymbol{a}$



(a) #18 POINT

(b) #4 BOW

Figure 4.3: The pose sequences and their histograms of (a) a pointing gesture and (b) a bowing gesture, respectively

the total number of K-means clusters used for (a). Note that when retrieving similar motions using (a) and (d), we select only one motion which has the best similarity by Equation 3.9 among overlapped time windows.

4.1.3 Classification Performance

For motion classification, we used an one-hour motion database in which we appended 30-minutes motions from five different subjects to the previous 30-minutes motions from the 10 subjects mentioned in Section 4.1.1. We extracted nine query gestures from the motion database. Table 4.3 describes the movement of the each gesture.

First, we compared the computational time for constructing the motion database with the four methods. Figure 4.4 shows the result of computational time [1]. The most time-consuming part is the wavelet transform in (a), the DP matching in (b) and (c), and the K-means clustering in (d). Because the time complexity of the DP matching is $O(N^2)$ for the number of motions N, the computational time of (b) and (c) was much longer than the others. In contrast, the time complexity of the *Bag-of-features*' approach is $O(N)$. Therefore, the time of (a) and (d) was drastically shortened and we can easily apply the proposed method to a motion database of larger scale.

Figure 4.6 shows the result of the motion classification. The vertical axis indicates the number of the similar motions classified. The number is dependent on a threshold for the similarity. In this experiment, we manually adjusted the threshold which a false positive motion was not classified for each gesture classification. According to the Figure 4.6, (a) the proposed method has the best classification performance. Since (c)

Table 4.1: Comparative experiments for motion classification. These methods indicate (a) the *Bag-of-motion-features*, (b) the *Match Web*, and (d) a method proposed by Raptis *et al*.

	Motion feature $\boldsymbol{f}$	Coordinate data $\boldsymbol{p}$
Histogram matching	(a)	(d)
DP matching	(c)	(b)

[1] We used a computer which has a Core i7 920 (2.66 [GHz] eight cores) and 16 [GB] memories.

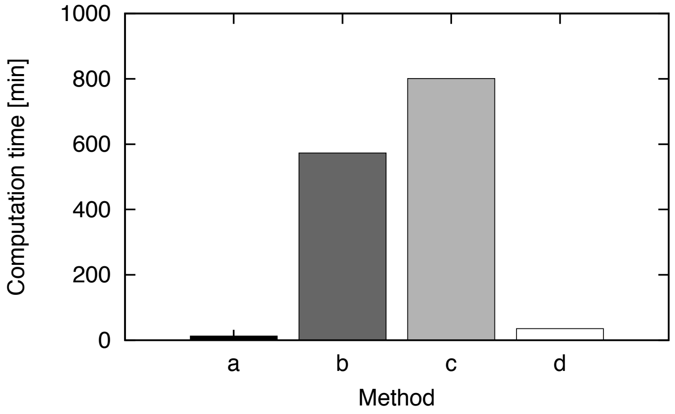


Figure 4.4: The computational time for constructing the motion database with the methods (a) to (d), respectively

Table 4.2: The actual computational times of Figure 4.4

method	time [min]
(a)	12.5
(b)	572.9
(c)	801.1
(d)	35.3

Table 4.3: Nine target gestures for the classification (a part of Table 3.2)

gesture		movement
#2	YES (NOD)	Nod several times
#4	BOW	Bow
#6	THINK	Fold both arms
#9	BYEBYE	Wave hand several times
#11	SWING	Swing hand
#12	SWING_BOTH	Swing both hands
#15	SHAKE_BOTH	Shake both arms quickly
#16	SPREAD_BOTH	Spread both arms widely
#18	POINT	Point at something

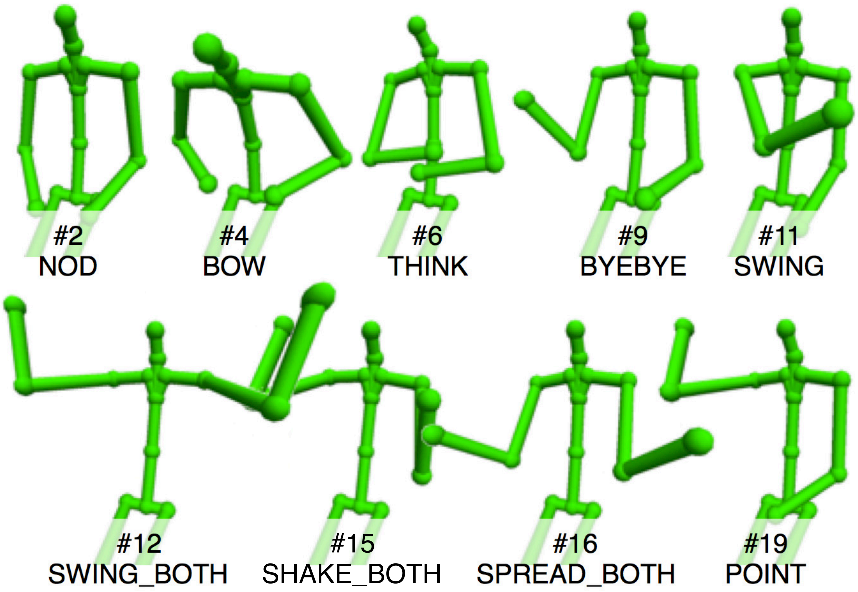


Figure 4.5: Nine target gestures for the classification

has almost the second best performance, it is effective to express motions in frequency domain. In addition, (a) is better than (d), and (c) is better than (b). That is, the *Bag-of-features*' approach is helpful for motion retrieval as well as image retrieval.

Next, we compared with ground truth of the number of similar motions. The ground truth is manually counted by an author of this thesis. Figure 4.7 shows the proportion of the number of the classified motions to the true number of the motions for each comparative method. Since the bowing gesture and nodding gesture have less individual variation in the movement than the other gestures, their proportion were quite higher. In contrast, the pointing gesture, folding arms gesture, and shaking gesture have more individual variation, their proportion was a little lower than the other gestures. However, the proposed method (a) had over 60% proportions for all gestures and drastically improved the classification performance compared with the *Match Web* (b). It is difficult to directly compare the results to the other related work, because they used different motion databases. However, for examples of body gesture recognition, Song *et al*. [71] measured 24 *NATOPS Aircraft Handling Signals* [1] with a stereo camera and classified motions by a *Particle Filter* [72] based method. Chen *et al*. [73] captured

[1] http://groups.csail.mit.edu/mug/natops/

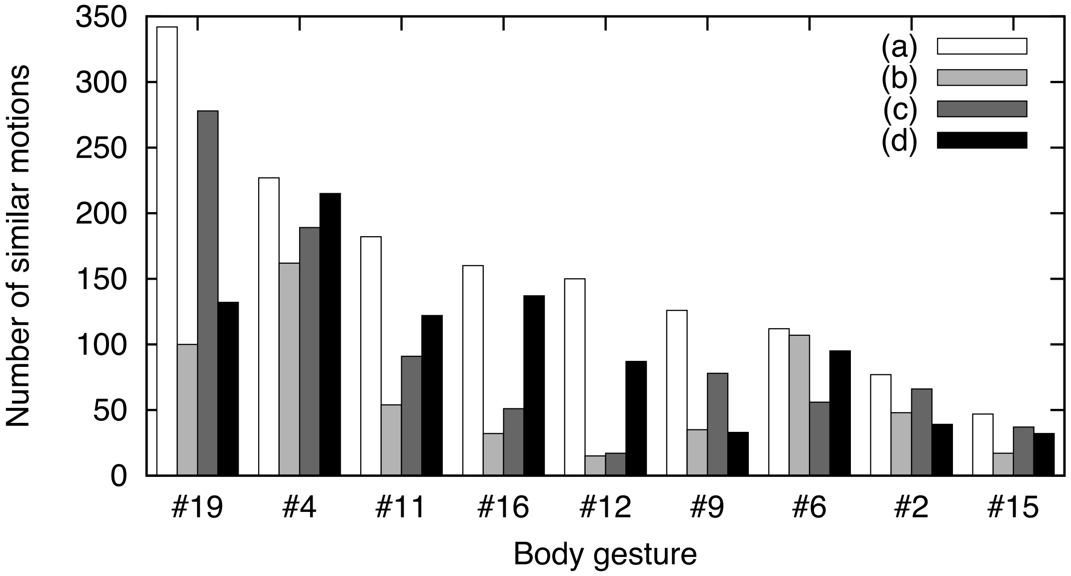


Figure 4.6: The number of similar motions of nine query gestures with the methods (a) to (d), respectively

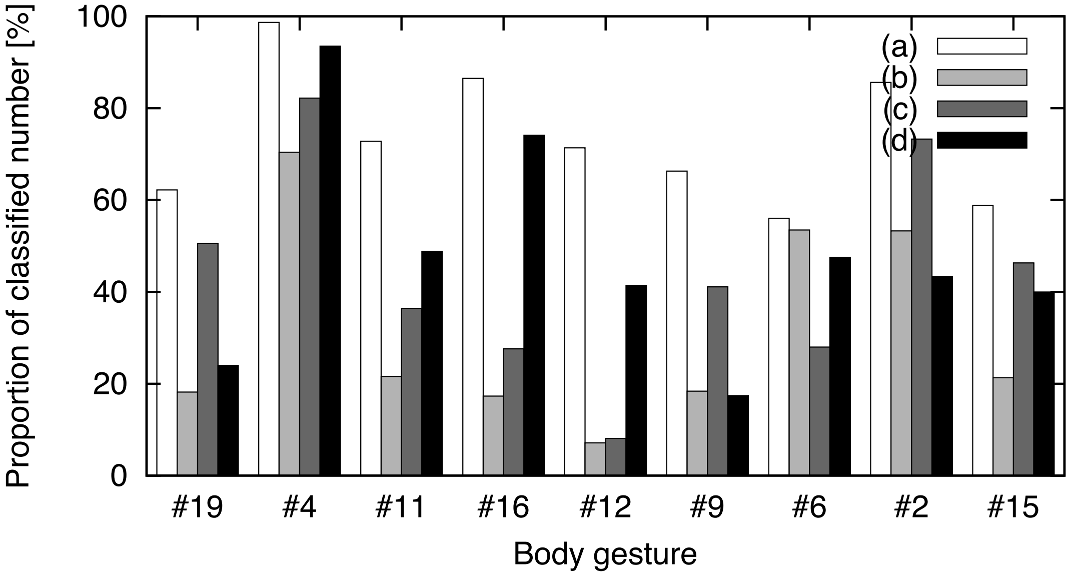


Figure 4.7: The proportion of the number of the classified similar motions to the true number of the similar motions with the methods (a) to (d), respectively

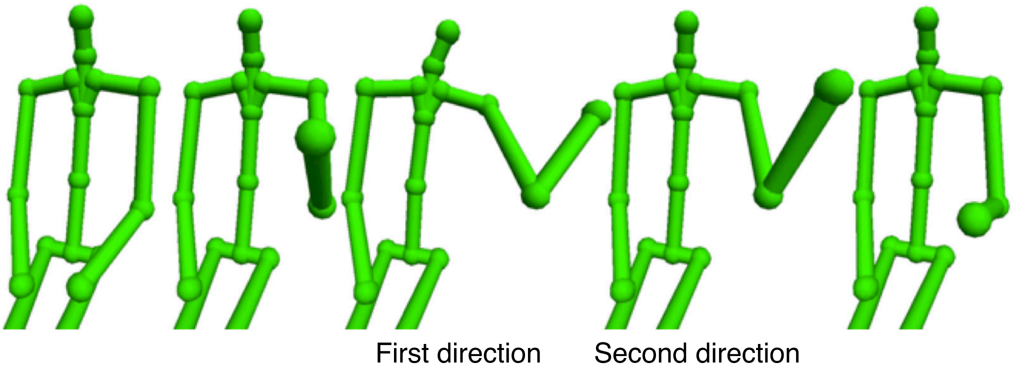


Figure 4.8: When a subject acted a pointing gesture, he pointed to two different directions at a time.

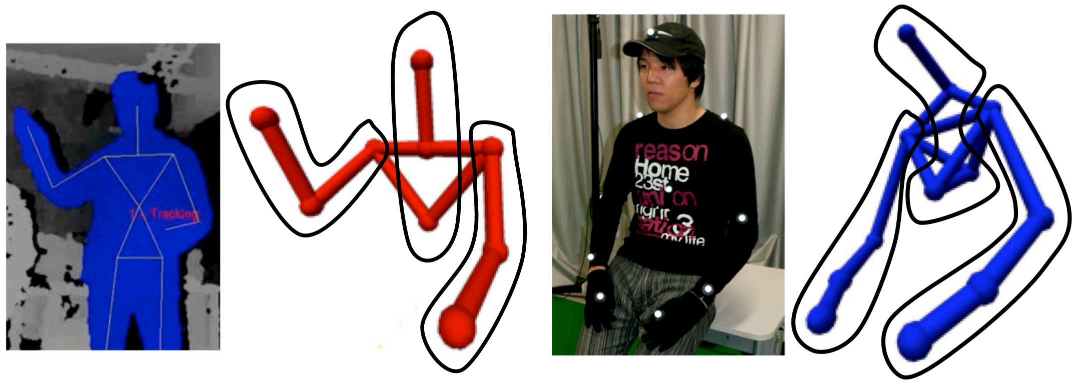
videos of 10 upper-body gestures and classified by a *Bag-of-features* based method. They had 60 to 70% and 65 to 70% of the recognition rates compared with ground truth, respectively. Therefore, we could say that our rates which have more than 60% are sufficiently large.

Discussion

Moreover, we discuss here in achievement of more numbers of the classification by the proposed method. During measurement of subjects' motions, subjects sometimes strongly acted distinctive gestures. For example, when a subject acted a pointing gesture which is the worst proportion in Figure 4.7, he pointed to two different directions at a time as shown in Figure 4.8. In this thesis, this gesture was counted as #18 POINT for the ground truth. However, the gesture's frequency was quite different from the target's one and it is also quite different among people how to count the ground truth. That is why the classification performance can be changed by how to count the ground truth and/or how to distinguish among gestures. In speech synthesis methods, they had also the same problem. We should discuss more how to make the ground truth in the near future.

4.1.4 Classification with Multi-Skeletons' Database

The proposed method, *Bag-of-motion-features* has one more capability which is a skeleton-independent classification. That is, *Bag-of-motion-features* can collect se-



(a) *KINECT*

(b) *MotionAnalysis*

Figure 4.9: The skeleton-models captured by (a) a RGB-D sensor *KINECT* and (b) an optical motion capture system, *MotionAnalysis*

mantic similar motions captured by different motion capture systems to construct a pre-designed motion database.

Three Motion Capture Systems

To evaluate the capability, we conducted a motion classification experiment using three different skeletons captured by three different motion capture systems. One of their systems is the sensor suit *MVN* which we used as shown in Figure 3.2. The other systems are shown in Figure 4.9 (a); a RGB-D sensor *KINECT* [1] made by Microsoft Corporation and Figure 4.9 (b) an optical motion capture system *MotionAnalysis* made by Motion Analysis Corporation. Figure 4.9 (a) and (b) are also shown how to divide their markers on upper body into three subsets: torso (s_{torso}), left arm (s_{left}), and right arm (s_{right}) as the same as *MVN* in Figure 3.2.

Since the *Match Web* cannot compute the similarity of motions which have different skeletal models, we conducted an experiment using only the proposed method.

Classification of Pointing Gesture

One subject's pointing gestures were captured by the *KINECT* and *MotionAnalysis*. Using these motions and the motion database captured by *MVN* mentioned in Sec-

[1] We use an open source library *OpenNI* to detect the human's skeleton.

tion 4.1.1, we conducted a classification experiment for #18 POINT.

Figure 4.10 shows the examples of truly classified pointing gestures in (a), (b) and (c) data sets, respectively. To compare with ground truth of the number of similar pointing motions, Figure 4.11 shows the proportion of the number of the classified motions to the true number of the motions. We used only one subject for conditions (b) and (c). That is why there were no individual variation in this experiment. Therefore, over 80% of motions were truly classified in their conditions compared to around 60% of motions in condition (a).

4.2 Motion Parameterization

We evaluate the precision of synthetic body gestures with respect to the target parameters. The precision is measured based on the number of database gestures (*i.e.* similar motions as mentioned above) per unit area. We conduct two experiments with the #18 POINT which has location parameters, and the #1 GAZE which has angle parameters.

4.2.1 Precision of Pointing Gesture

Figure 4.12 (a) shows the area of target locations for #18 POINT. The area is 500 [mm] × 500 [mm] square and defined as $(x, 400\ [mm], z)$ where $-100\ [mm] \leq x \leq 400\ [mm]$, $300\ [mm] \leq z \leq 800\ [mm]$. Given n database gestures which are placed at regular intervals in the area, the Actroid generates a gesture by synthesizing while changing the number n of the similar motions to synthesize. We experimented $n = 4, 5, 9, 16$ cases. For each experiment, 1000 target locations were sampled by uniform distribution within the area and synthetic gestures generated to satisfy the target parameters. In case of $n = 4, 9$, the results are shown in Figure 4.13. White circles in Figure 4.13 indicate the n database gesture's target, and plus dots indicate the synthetic gesture's position.

As shown in Figure 4.13, the larger number of n was used, the more uniform locations could be generated. In case of $n = 4$, the plus dots are dense around four corners. It causes less density of the central region in the target area. Figure 4.14 shows the result of mean errors between target location and synthetic gesture's location. The mean errors were monotonically decreasing in proportion to n. In usual HRI, small errors

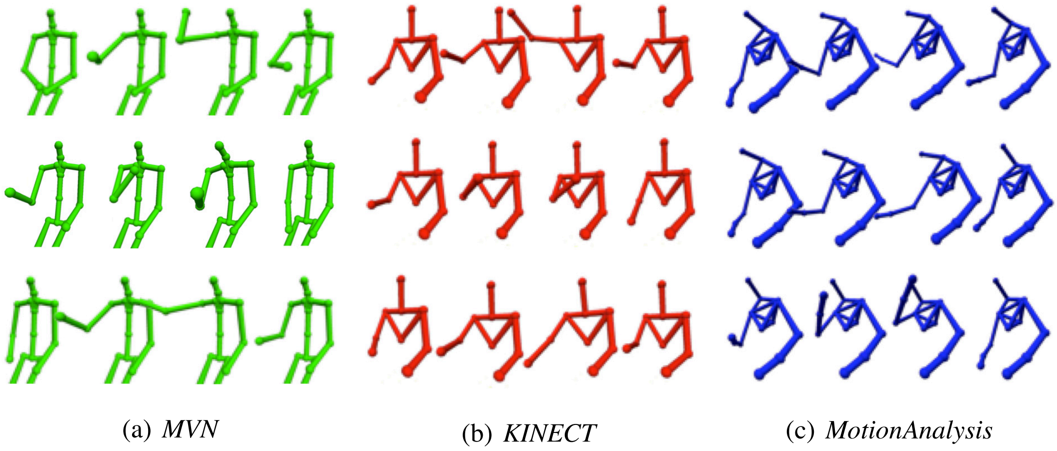


Figure 4.10: The three results of the classified pointing gestures in (a), (b) and (c) data sets, respectively

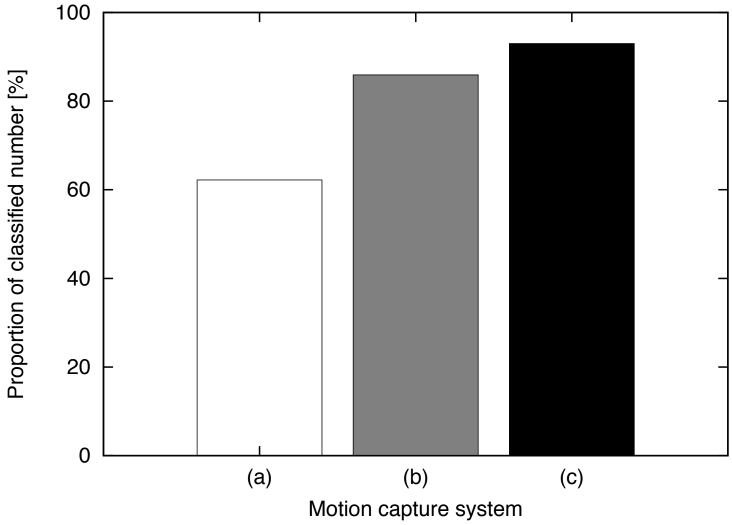
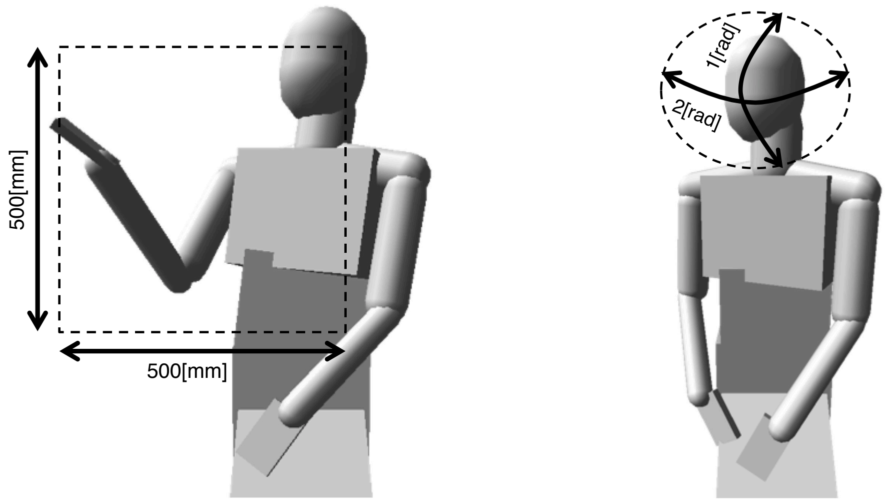


Figure 4.11: The proportion of the number of the classified pointing motions to the true number of the motions with the methods (a) to (c), respectively



(a) POINT

(b) GAZE

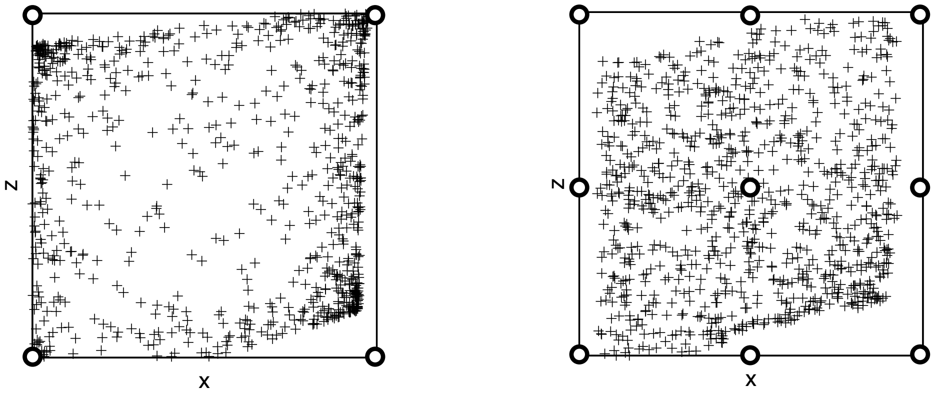
Figure 4.12: The definition of target parameters: (a) target location of #18 POINT and (b) target angle of #1 GAZE

in gesture are tolerable. If we tolerate about 100 [mm] errors, $n = 5$ (*i.e.* 250 [mm] sampling intervals) has enough accuracy for the pointing gesture.

4.2.2 Precision of Gazing Gesture

Gazing gesture is actually a static pose and not a motion sequence. However the Actroid achieves gaze gesture by smoothly transitioning from pose to pose, and by planning gaze movement based on the convergence of eyes and the sharing rate of the head direction and the eye direction. Figure 4.12 (b) shows the range of target angles for #1 GAZE. The range is defined as (ϕ, θ) where $|\phi| \leq 1.0[rad], |\theta| \leq 0.5[rad]$.

As in Section 4.2.1, given $n = 6, 8, 14$ sampling gestures which are placed at regular intervals in the range, the Actroid generates gestures by synthesizing the n samples. For each experiment, 1000 target angles were sampled by uniform distribution within the area and synthetic gestures were generated to satisfy the angle. In case of $n = 6, 8$ the results are shown in Figure 4.15, and Figure 4.16 shows the result of mean errors when $n = 6, 8, 14$. If we tolerate about 0.2[rad] (= 11.4[deg]) errors, $n = 6$ (*i.e.* 1.0[rad] sampling intervals) has enough accuracy for the gaze gesture.



(a) $n = 4$

(b) $n = 9$

Figure 4.13: Visualization of accessible pointing locations in xz plane: in case of (a) $n = 4$ and (b) $n = 9$, where white circles and plus dots indicate database gesture's target and synthetic gesture's target, respectively

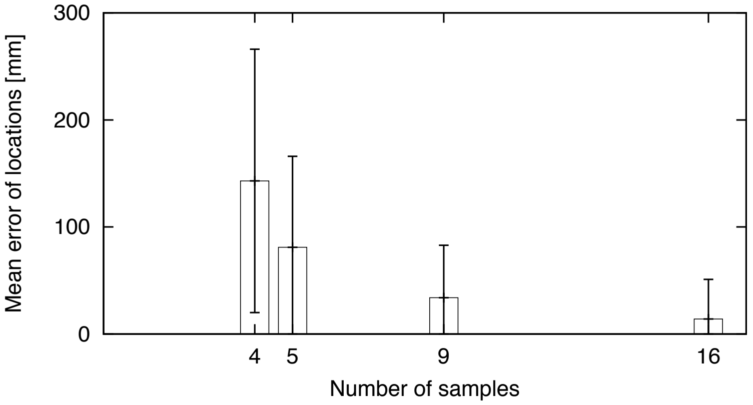
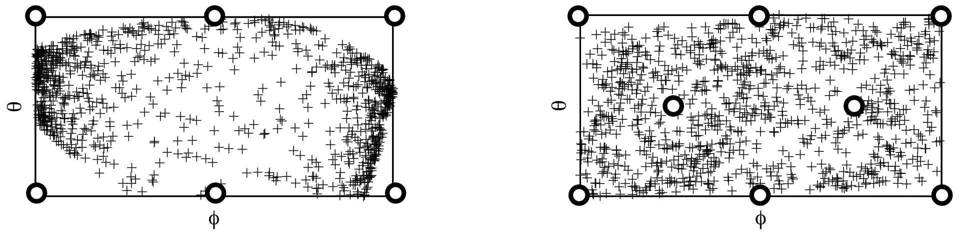


Figure 4.14: Mean errors between target location and synthetic pointing gesture's location in case of $n = 4, 5, 9, 16$



(a) $n = 6$

(b) $n = 8$

Figure 4.15: Visualization of accessible gaze angles in $\phi\theta$ plane: in case of (a) $n = 6$ and (b) $n = 8$, where white circles and plus dots indicate sampling gesture's target and synthetic gesture's target, respectively

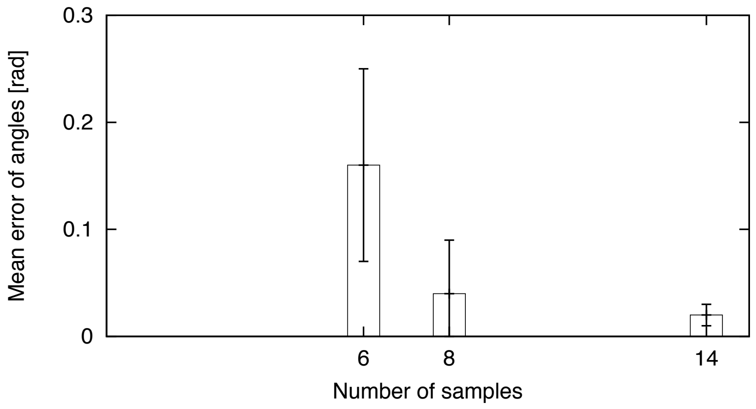


Figure 4.16: Mean errors between target angle and synthetic gaze gesture's angle in case of $n = 6, 8, 14$

4.2.3 Discussion

As mentioned above, in usual HRI, small errors are tolerable. Therefore we assumed to tolerate about 100 [mm] errors for the location and about 0.2[rad] errors for the angle. As a result of Section 4.2.1 and Section 4.2.2, we have the following guidance to design a gesture.

- A gesture which has (x, y, z) position parameters:
  The sampling resolution should be less than 250 [mm] intervals in a target area.
- A gesture which has (ϕ, θ) angle parameters:
  The sampling resolution should be less than 1.0[rad] intervals in a target range.

4.3 Motion Interruption

4.3.1 Effectiveness of Self-Collision Avoidance

First, we evaluate the effectiveness of the self-collision avoidance as mentioned in Section 3.5.4. Figure 4.17 shows the comparison between the self-collision avoidance enabled/disabled with trajectories of both hands in a collision case. In the disabled case, collision occurred at about 0.4 [s] and the hand's velocity was changed oscillatory. In the enabled (*i.e.,* the proposed method), there is no collision and two motions could be connected smoothly.

4.3.2 Evaluation of Velocity Information

Next, we evaluate performance about the connection smoothness by comparing the proposed method with the following three approaches.

- *Direct Connection*
  Connect two motions directly at an interruption time without any interpolation.
- *Linear Interpolation*
  Connect two motions by straight-line interpolation at an interruption time.
- *Motion Graph*
  Connect two motions using Equation (3.11) at an interruption time even if the distance is larger.

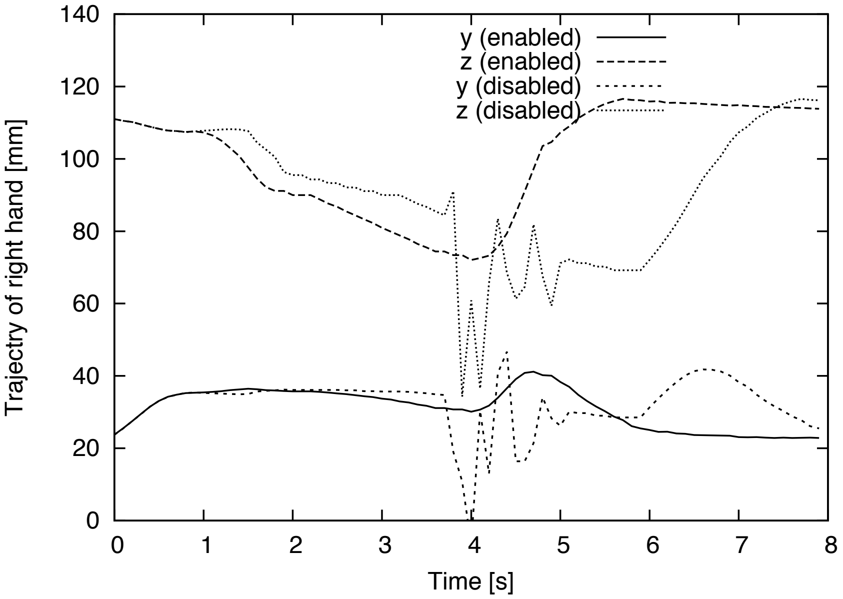


Figure 4.17: Comparison between the self-collision avoidance enabled/disabled with trajectories of both hands when experimented a collision case

We used motions #14 SHAKE_BOTH, #7 FLEX_MUSCLE, and #16 SPREAD_BOTH in a row as shown in Figure 4.18. We chose these motions to verify the collision avoidance; connection between the two motion tends to collide both hands. In these simulations, we performed 264 trials where transition points are changed at 0.33 [s] intervals within each motion.

Figure 4.21 shows the averages of hand's velocity where frames of the connecting area and its few back and front one. The error bars mean standard deviation that can be interpreted as acceleration. In both Direct Connection and Motion Graph, higher velocity and acceleration appear (*i.e.,* it means couldn't connect smoothly each other).

4.3.3 Evaluation of Key-Pose Information

Next, we verified the smoothness of connected motion in real robot. Retroreflective markers are attached to the Actroid-SIT as shown in Figure 3.1 (a), and the smoothness is analyzed quantitatively using motion capture system.

In this experiments, #7 FLEX_MUSCLE and #15 SPREAD_BOTH in a row shown in Figure 4.18 are utilized. Figure 4.22 shows y and z values of both hands. Dot-lines

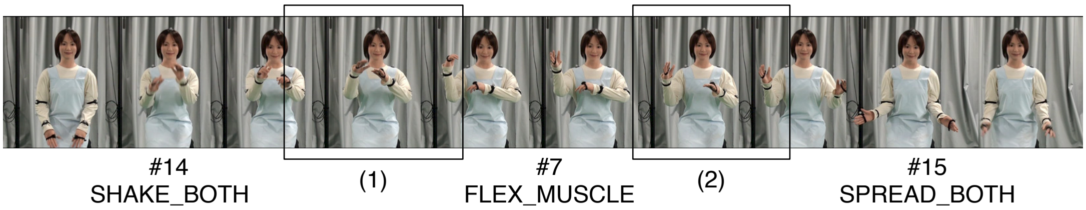
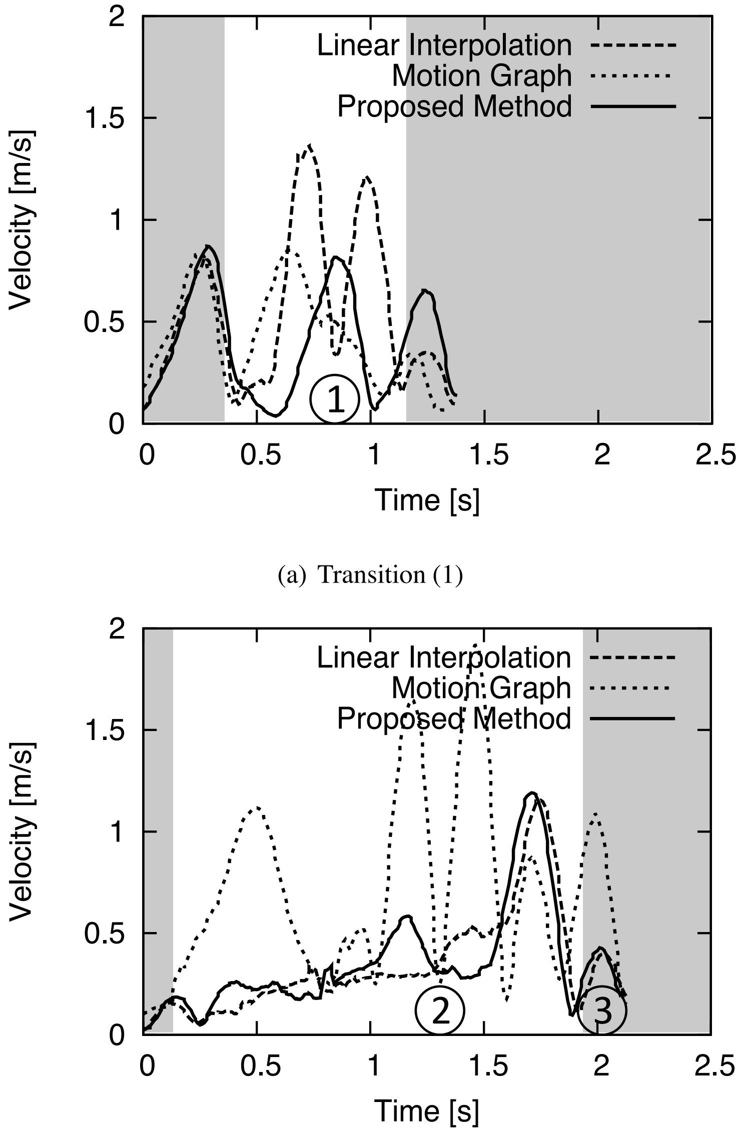


Figure 4.18: A series of three gestures connected by *Proposed Method*, where square windows show the transition durations

indicate the first key pose timing of the motion. Down-arrows indicate the interruption timing to shift the next motion.

Figure 4.22 (1) shows the result of interruption at the end of motion #7 to see the true value of key-pose information of each gesture. We compared key pose values with Figure 4.22 (2) and Figure 4.22 (3) based on this result.

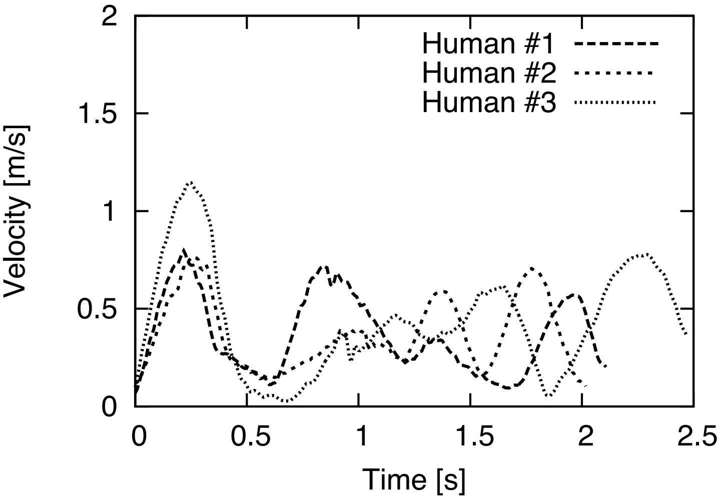
There are some loss of key-pose information where the second interruption and the first interruption in Figure 4.22 (2) and Figure 4.22 (3), respectively. However, with the total data of the whole Actroid markers, we confirmed not to lose the key-pose information. The exception of that, a loss of the key-pose information and a sharp velocity changes associated with the self-collision have nothing. It seems to be maintained their motion features.



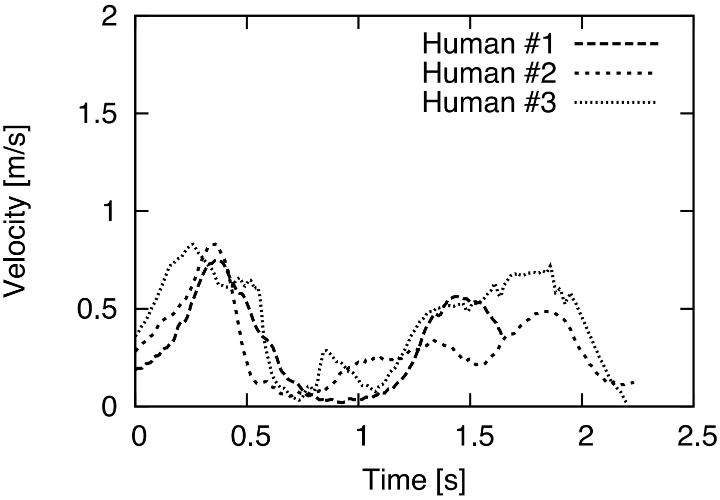
(a) Transition (1)

(b) Transition (2)

Figure 4.19: The right hand's speed during transition durations of Figure 4.18 (1) and (2)

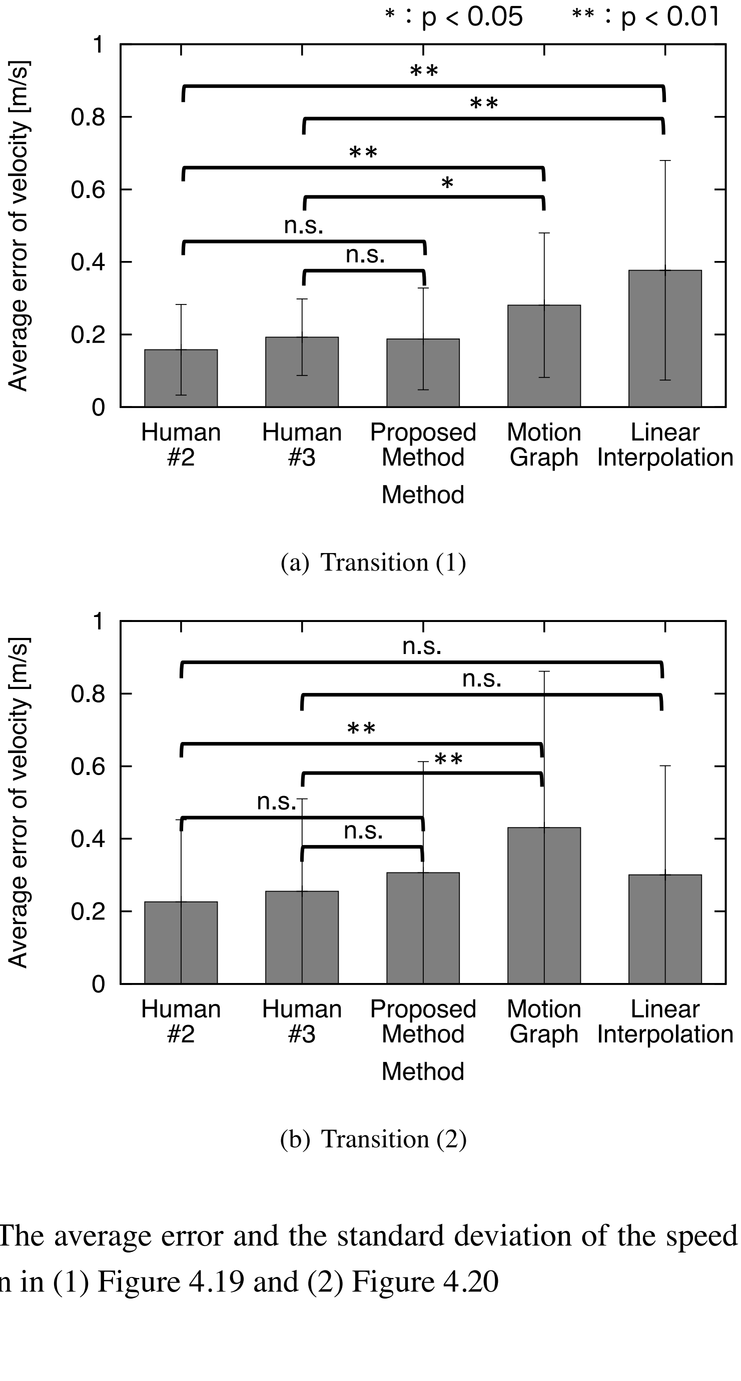


(a) Transition (1)



(b) Transition (2)

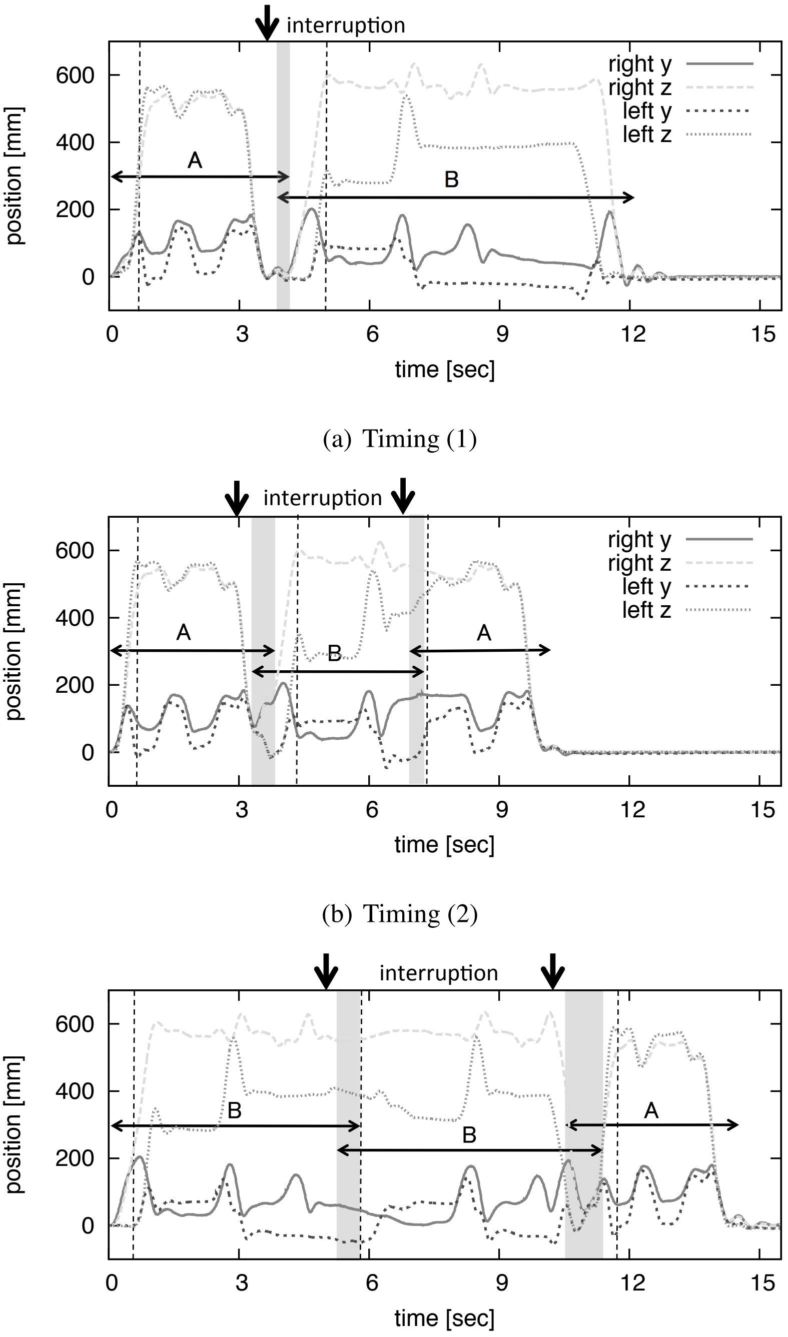
Figure 4.20: The right hand's speed of three people during the same gestures as in Figure 4.18



(a) Transition (1)

(b) Transition (2)

Figure 4.21: The average error and the standard deviation of the speed during transitions as shown in (1) Figure 4.19 and (2) Figure 4.20



(a) Timing (1)

(b) Timing (2)

(c) Timing (3)

Figure 4.22: The result of motion transitions whose initial position are 0 [mm] with interrupting at different three timings (1), (2) and (3) where gray zone indicates connecting motion area

Chapter 5

Multi-Party Human-Robot Interaction System

5.1 Concept

Communication with multiple people is more common than one-to-one communication. We therefore develop the system for multi-person communication. By embedding the RMDB, we propose gesture adjustment suitable for human's demand through parameterization and gaze movement planning which can communicate with multiple people and adjust a gesture to the location of talker and/or object. We implemented the HRI system on the Actroid.

5.2 System Configuration

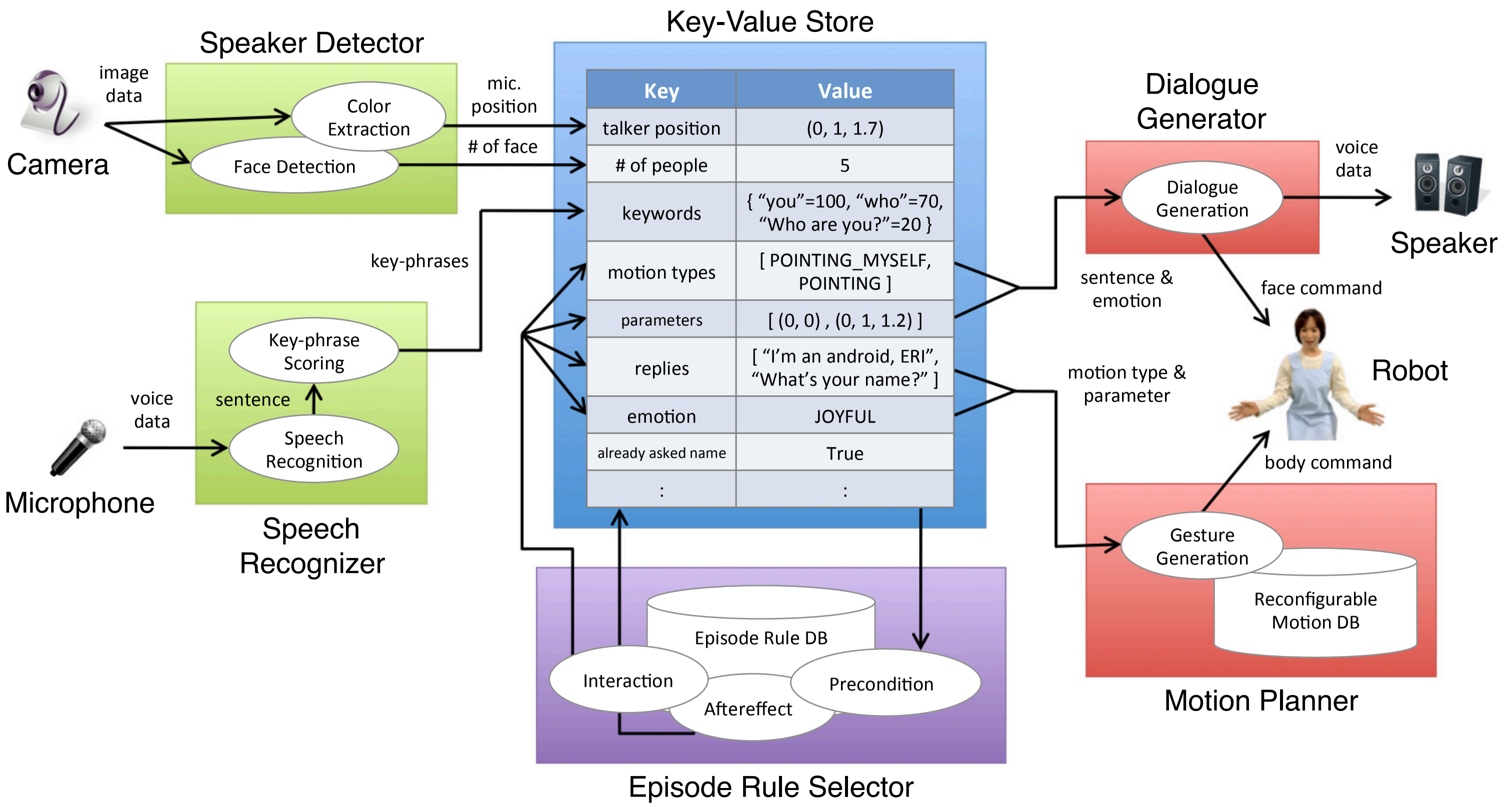
Figure 5.1 shows the system configuration of our proposed system. This system has six components: the *Key-Value Store*, the *Episode Rule Selector*, the *Speaker Detector*, the *Speech Recognizer*, the *Motion Planner*, and the *Dialogue Generator*.

5.3 System Components

5.3.1 Key-Value Store

The *Key-Value Store* is one of the dictionary data structures that have a set of pairs, which include the index (*Key*) and the related value (*Value*). The *Key-Value Store* has been developed as an alternative to the *Relational Database* [74] which is inferior due to an overhead of network communication and the difficulty of parallel processing. Since the *Key-Value Store* uses a distributed hash table, it is easy to process in parallel with redundancy. We use *Redis* [1] which is one of the *Key-Value Store* implementations, and let it perform as a web server.

[1] http://redis.io/



Key	Value
talker position	(0, 1, 1.7)
# of people	5
keywords	{ “you”=100, “who”=70, “Who are you?”=20 }
motion types	[POINTING_MYSELF, POINTING]
parameters	[(0, 0) , (0, 1, 1.2)]
replies	[“I’m an android, ERI”, “What’s your name?”]
emotion	JOYFUL
already asked name	True
:	:

Figure 5.1: System configuration of the multi-party HRI system which has six components: the *Speaker Detector* and the *Speech Recognizer* are for sensing; the *Dialogue Generator* and the *Motion Planner* are for the Actroid, and the *Episode Rule Selector* is for planning how the Actroid should interact. All components communicate via the *Key-Value Store*.

All components in the HRI system access this *Key-Value Store* using a *RESTful API* [75]. This API is thread-safe and allows dynamic changes of connection among modules. Various architectures have been proposed for robot systems, and stability and flexibility have been studied in these architectures. In recent architectures, *OpenRTM-aist* [76] and *ROS* [77] have become popular. Since they provide operating system-like functionality, they are heavy and inconvenient to use only for connections and communication among the components of the system. Therefore, we applied the *Key-Value Store* to the robot system as a centralized administrative framework.

5.3.2 Episode Rule Selector

The *Episode Rule Selector* decides the reaction of the Actroid based on the episode rule database when sensory data changes. Kanda *et al.* [53] originally proposed the idea of the *Episode Rule Selector*. In this research, the episode rules are described using a script language, *Jython* [1]. Therefore, the rules can be modified and added online even while the system is running. Each episode rule contains the following three functions.

- *Precondition*
  Return a score calculated based on the speaker's dialogue, existence of the speaker, and the number of people, etc., which are obtained from the *Key-Value Store*.

- *Interaction*
  Configure a gesture's type and its parameter, and a reply's words and emotions. In the case of Figure 5.1, the Actroid replies: "What's your name?" with two gestures to promote the next interaction.

- *Posteffect*
  Terminate this rule, and set or reset several condition flags in the *Key-Value Store* for the next interaction. The key "already asked name" can be seen in Figure 5.1. This key is the flag for the next continuous interaction "My name is ...", which we expected.

The algorithm of *Episode Rule Selector* is shown in Figure 5.2. While the system is running, the algorithm repeats the following procedures. First, the episode

[1] http://www.jython.org/

```
while system is running do
    $\mathscr{R} \leftarrow \{\mathscr{R} \cup \text{updated or added rules}\}$
    $r = \arg\max_{r_i \in \mathscr{R}} r_i.precondition.$
    $r.interaction$
    $r.posteffect$
end while
```

Figure 5.2: The algorithm of the *Episode Rule Selector* implemented on the proposed HRI system. The terms $\mathscr{R}$ indicates the episode rule database and r indicates the rule which the system selected.

rule database $\mathscr{R}$ is loaded. Then, the system selects the episode rule r which has the maximum score of the $precondition$ in $\mathscr{R}$, and the functions of $r.interaction$ and $r.posteffect$ are called in order. This interaction loop can be interrupted even if a previous $r.interaction$ is not finished. Current gesture can be transitioned to the next gesture smoothly thanks to motion interruption, and the *VoiceText* can interrupt current voice and resume with the next voice. By planning the interruption-aware interaction, we accelerate reaction speed.

Our system can deal with action episode rules which show that the robot actively performs an interaction when there is no speaker, as well as reaction episode rules which are passive rules when the speaker asks a question. Appendix C describes how to make the episode rule by *Jython*.

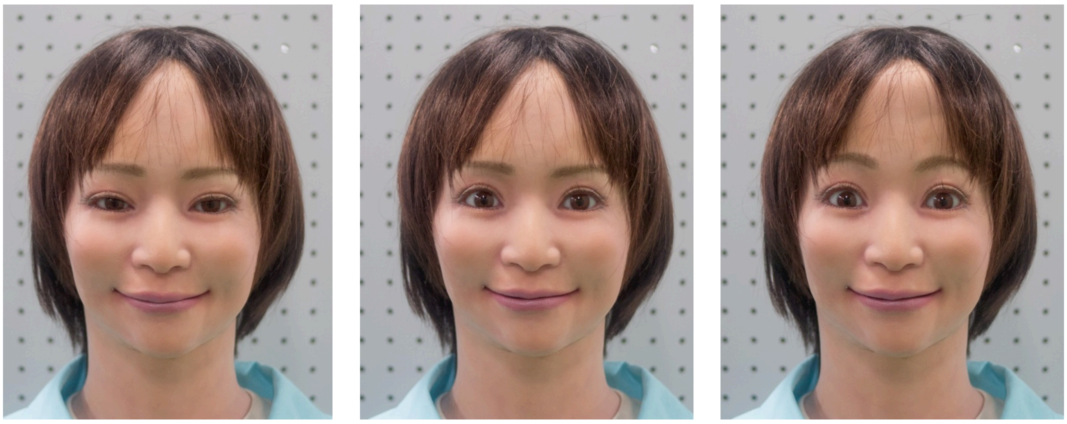
For example, one of the passive rules is to receive question about the Actroid's age. It has a *Precondition* which matches speaker's keywords to a set {"how", "old", "you", "built", "made"}, an *Interaction* which makes two gestures POINT_AT_MYSELF and POINT based on speaker's location and one dialogue "I am three years old, and you?", and an *Posteffect* set a flag "already asked age."

5.3.3 Motion Planner

This *Motion Planner* have already been described in Section 3.4 and Section 3.5.

5.3.4 Dialogue Generator

The Actroid can control facial expression and gaze direction in the same way as a human. Retargeting the human's facial expression into the Actroid's facial expression



(a) SAD/ANGRY (b) NORMAL (c) JOYFUL/HAPPY

Figure 5.3: Comparison of facial expressions: (a) SAD/ANGRY, (b) NORMAL, and (c) JOYFUL/HAPPY. The Actroid can control only the eyebrow, eyelid and cheek for facial expressions.

is very difficult because of the complexity of the actuator. Therefore, in our system, the module to control facial expression and gaze is implemented independently with the RMDB. In addition, we describe speech generation here.

Since the Actroid has the eyebrow, eyelid and cheek joints for controlling facial expression, we use the Actroid to perform simple facial expression as shown in Figure 5.3. The Actroid has five emotions: NORMAL, SAD/ANGRY, and JOYFUL/HAPPY. To distinguish between SAD/ANGRY, and JOYFUL/HAPPY, we also change voice parameters (*i.e.,* speed, pitch, volume, and pause time) to make the expressions unique using a voice synthesis software, *VoiceText* [1] made by HOYA Service Corporation.

5.3.5 Speaker Detector

To detect the speaker's position and to count the number of people around the Actroid, we use an IEEE 1394 camera (Sony DFW-VL500) located next to the Actroid. Figure 5.4 shows the result of the image processing. Circles in this figure indicate the

[1] http://voicetext.jp/

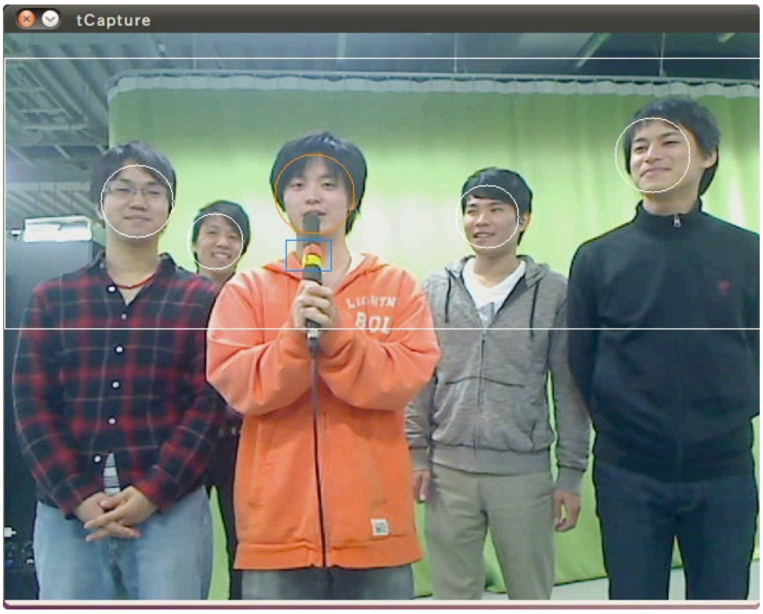


Figure 5.4: An example of the image processing, where circles indicate the results of face detection and the blue square indicates the result of color extraction for microphone detection

result of face detection by Haar-like features [78] in the *OpenCV* [2] library. A blue square also indicates the result of a color extraction of tapes attached on the microphone. It is difficult to detect a speaker (*i.e.,* the person who is speaking) only with face information. Therefore we assumed a person who has a microphone is the speaker. We used two different colors (yellow and blue) for detection.

5.3.6 Speech Recognizer

Julius [1] is employed as speech recognition software, and we employ the *Yahoo Keyphrase Extraction Web API* [2] to extract important keywords, which are a subset of the sentence recognized by Julius. The keyword sometimes consists of multiple words. The API scores each keyword based on its importance. In Figure 5.1, the sentence: "Who are you?" is divided into the keywords "you", "who", and the sentence: "Who

[2] http://sourceforge.net/projects/opencvlibrary/

[1] http://julius.sourceforge.jp/

[2] http://developer.yahoo.co.jp/webapi/jlp/keyphrase/v1/extract.html

are you?”. For robust matching, the whole sentence is added as a low priority keyword.

In Japanese especially, it is very difficult to grammatically recognize the speaker’s speech. However, this system employ word spotting approach with scored keywords. For example, in case of replying to: “What is your name?”, Table 5.1 shows four answers which mean almost the same: “My name is Kondo.”, but their grammar and the way of speaking are different. However, their primary keywords are the speaker’s name “Kondo” in all cases, so that the Actroid can reply: “Your name is Kondo, isn’t it?”.

Table 5.1: Sentences and the sets of their keyword and score for replying to “Who are you?” Higher score indicates the keyword is more important. 100 is the highest score and 0 is the lowest score.

sentence	the set of keyword and score
My name is Kondo.	{“Kondo”=100, “name”=46, “my”=21}
Please call me Kondo.	{“Kondo”=100, “call”=39, “me”=21}
I am Kondo.	{“Kondo”=100, “I”=21}
This is Kondo.	{“Kondo”=100, “this”=18}
Kondo.	{“Kondo”=100}

Chapter 6

Subject Experiments of Multi-Party Interaction

To evaluate the effectiveness of our proposed system, we conduct a lot of subject experiments. These experiments contain not only quantitative evaluations such as the measurement of response/residence time of communication but also qualitative evaluations such as the analysis of human impressions of the Actroid.

6.1 Response Time

First, to verify the effectiveness of interruptivity, conversations between 68 speakers and the robot were conducted by *interruptible* (*i.e.,* the interruption is enabled) and *un-interruptible* (*i.e.,* the interruption is disabled).

6.1.1 Method

Table 6.1 shows the number of subjects and the number of communications; one communication is defined as the period when a human talks to a robot and the robot reply. In Table 6.1, α is the number of communications where a human interrupts the previous interaction and β is the number of communications where a human didn't interrupt. It is desirable for HRI that the response time in case of both α and β is equivalent each other. Interruption due to failure of speech recognition is not distinctive. When the

Table 6.1: The number of subjects and communications

		un-interruptible	*interruptible*
# of subjects		30	38
# of communications	α	29	48
	β	98	130

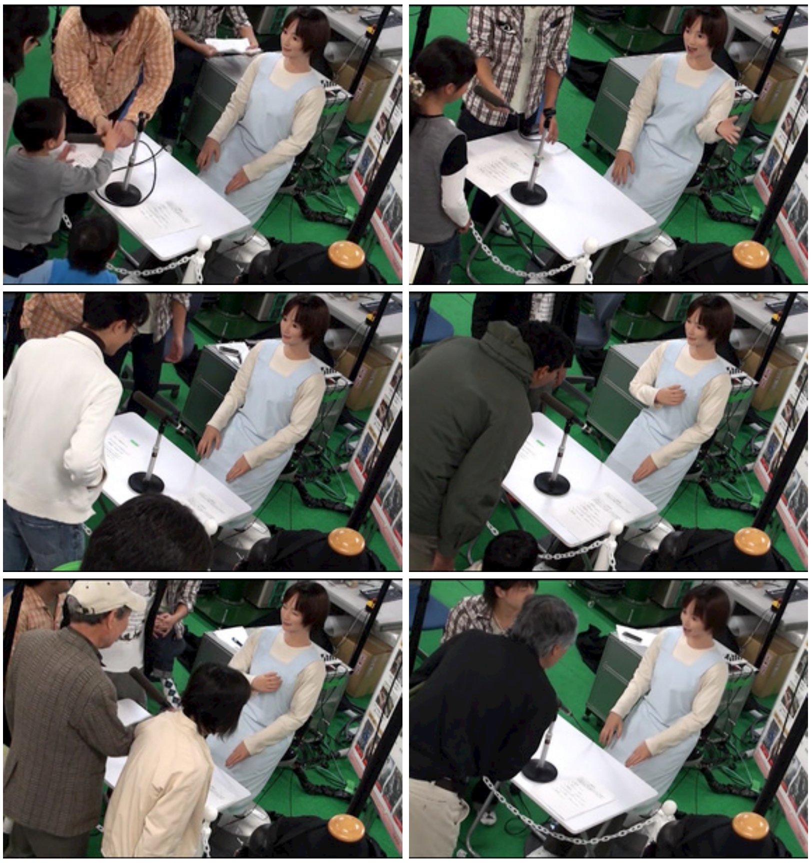


Figure 6.1: The snapshots of HRI experiments whose subjects are in various age groups

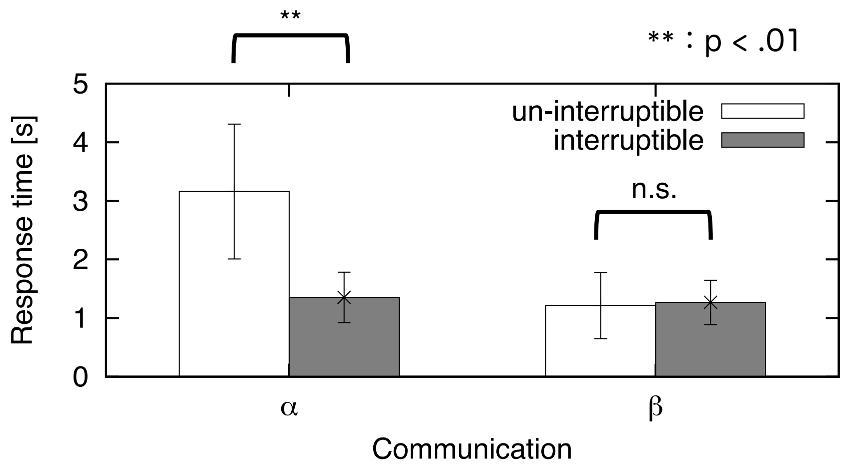


Figure 6.2: Comparison of the response time between *un-interruptible* and *interruptible*

HRI system cannot decide the reaction from result of speech recognition failure, the robot reacts for the request ”Can you say it one more time, please?”

Experiments were conducted for visitors who are in various age groups. Figure 6.1 shows the snapshots of experiments.

6.1.2 Results

Figure 6.2 shows average and standard deviation of response time. The response time is defined as the duration from the moment which subject finishes speaking to the moment which Actroid-SIT starts executing the gesture. The main causes of the deviation, are the difference of gesture length and whether success or failure of recognition by Julius (*i.e.,* the failure case is required more calculation time).

First, if the subject did not interrupt during the robot communicates, *interruptible* and *un-interruptible* have the same algorithm. That’s why, they have little difference as seeing the data which people did not interrupt in Figure 6.2. On the other hands, they have the significant difference with the data which people attempted to interrupt. While with *un-interruptible* the response time is twice slower, with *interruptible* it is almost same time as in non-interruption case. Therefore, we confirmed reducing response time. That is, our proposed method, *interruptible* can make HRI more smoothly.

6.1.3 Discussion

Through whole experiments, younger and elder person tend to interrupt the previous interaction more than adults do [1]. Adults understand that robots are immature and imperfect in communication and thus we can see their behavior to wait the incorrect reaction without interruption. However, we cannot see such behavior in younger or older person. The proposed HRI system is more effective for them.

6.2 Speaker Ratio and Residence Time

We evaluate the effectiveness of the proposed method, *i.e.,* motion interruptivity and motion parameterization. To do that, we compare the proposed system to the systems where motion interruption and/or motion parameterization are not implemented.

6.2.1 Hypotheses and Predictions

The proposed motion interruption and parameterization make an HRI system more responsive and active. We believe people feel comfortable to communicate through the system, because the motion parameterization makes the Actroid active thanks to gestures which has ϕ, θ parameters such as idling motion by gesture #1 GAZE. This is why the Actroid is busy looking around at people, and subjects might feel easy talking to the Actroid. In addition, the motion interruption makes for responsive communication as mentioned in Section 6.1. Thus, the response time (*i.e.,* the latency of communication) is decreased nevertheless the communication can be lively and durable. We expect that a better HRI system gets higher speaker ratio and longer residence time. Hereby, we made the following predictions.

- *Prediction 1*
  The motion parameterization will increase people who voluntarily speak to the Actroid.
- *Prediction 2*
  The motion interruption will increase the residence time with the Actroid.

[1] Experiment were conducted for visitors, thus we couldn't collect their true age information. Each person is divided into three large groups manually, so there is ambiguity in these data. But we can confirm the tendency of communication depending on age.

6.2.2 Conditions

We controlled two conditions, the motion interruption (*interruptible*) and the motion parameterization (*parametric*). Combining these conditions, there are four comparative systems, **UI+NP**, **I+NP**, **UI+P**, and **I+P**. The systems are shown in Table 6.2. Each system is indicated as follows:

- **UI+NP** is a conventional interaction system which does not consider both motion interruption and parameterization.
- **I+NP** is the one-to-one interaction system which enabled only the motion interruption.
- **UI+P** can parameterize gestures but cannot interrupt interaction.
- **I+P** is the multi-party interaction system which we proposed.

Of course the systems **UI+NP**, **I+NP**, and **UI+P** can interact with multiple people, but **UI+NP** and **UI+P** cannot interrupt the interaction and, **UI+NP** and **I+NP** can interact with people assuming that they are in front of the Actroid.

6.2.3 Method

Participants

Experiments were conducted for four weekdays in the Heijo palace site, Japan. One of the above-mentioned systems was evaluated on one of the weekdays. 1,662 visitors in total took part in these experiments as the subjects. Table 6.2 shows the number of the speakers, non-speakers, and their totals (*i.e.,* subjects) for each day. The *subject* in the table is the number of people who approach the Actroid, the *speaker* is the number of subjects who voluntarily spoke to the Actroid, and the *non-speaker* is the number of subjects who did not speak to the Actroid (*i.e., subject* = *speaker* + *non-speaker*). Subjects were in various age groups and had a similar age distribution for each day.

Settings

We used the Actroid as a receptionist and used Q & A communication, such as Q: "Where is the bathroom?" A: "It is on your right." It can also communicate dozens

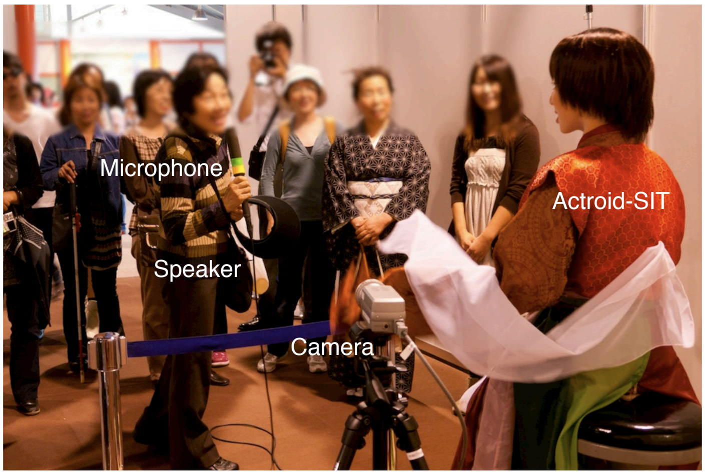


Figure 6.3: Snapshot of HRI experiments when a speaker asks the Actroid-SIT

Table 6.2: The number of subjects, speakers and non-speakers with respect to four comparative systems. The terms *interruptible*/*un-interruptible* means the motion interruptivity was enabled/disabled and *parametric*/*non-parametric* means the motion parameterization was enabled/disabled.

system	condition	*speaker*	*non-speaker*	*subject*
UI+NP	*un-interruptible* and *non-parametric*	226	241	467
I+NP	*interruptible* and *non-parametric*	165	267	432
UI+P	*un-interruptible* and *parametric*	263	166	429
I+P	*interruptible* and *parametric*	206	128	334

of daily conversations such as Q: "How old are you?" A: "I was born three years ago. And you?" Figure 6.3 shows a snapshot of the communication when a speaker asked a question to the Actroid.

We assumed that among the subjects, only one subject would be allowed to speak to the Actroid at one time. This assumption is plausible because it is difficult, even for humans, to answer questions asked by multiple people simultaneously. We regard a single person as a special case of multi-party HRI; we do not control the number of attendees at one time.

6.2.4 Measurement

Two measurements were conducted:

· *Speaker ratio*
Define as the ratio of subject who spoke to the Actroid (i.e, $\frac{speaker}{subject}$).

· *Residence time*
Define as the duration from the moment when the subject starts speaking to the moment when the subject puts his/her microphone back onto its stand [1].

6.2.5 Results

First, we verified the *Prediction 1*. Figure 6.4 shows the results of *speaker* and *non-speaker* for each system. A chi-square test was revealed significant differences among conditions ($\chi^2(3) = 62.765, p < .01, \phi = 0.194$). A residual analysis revealed that *speaker* in **UI+NP** is significantly low (residual= $-1.79, p < .10$) and *non-speaker* in **UI+NP** is significantly high (residual= $1.79, p < .10$). *speaker* in **I+NP** is significantly low (residual= $-6.552, p < .01$) and *non-speaker* is significantly high (residual= $6.552, p < .01$) in **I+NP**. *speaker* in **UI+P** is significantly high (residual= $4.064, p < .01$) and *non-speaker* is significantly low (residual= $-4.064, p < .01$) in **UI+P**. *speaker* in **I+P** are significantly high (residual= $4.601, p < .01$) and *non-speaker* in **I+P** are significantly low (residual= $-4.601, p < .01$). These results implicitly indicates the motion parameterization increases *speaker ratio*.

[1] We assumed that in many cases, putting the microphone back represents to lose interest in the Actroid.

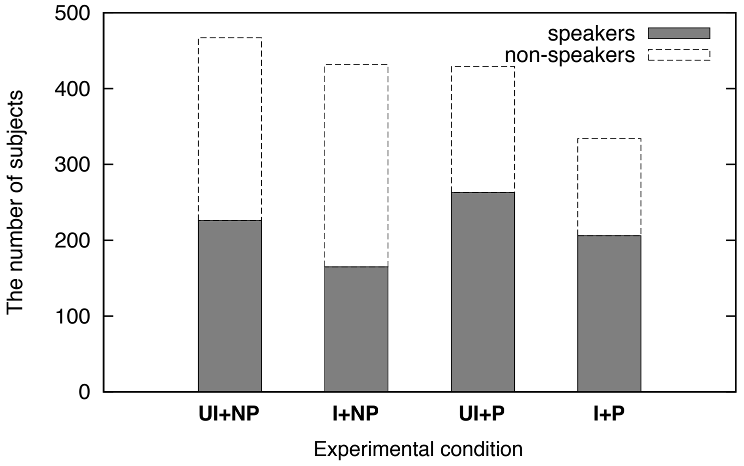


Figure 6.4: Comparison of *speaker* and *non-speaker* between four experimental systems

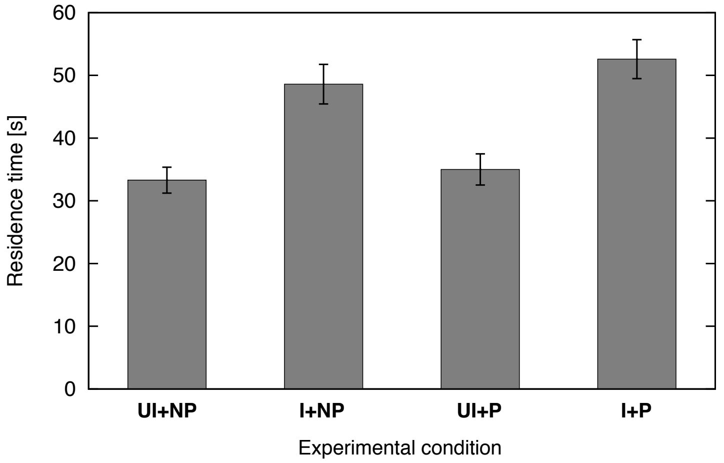


Figure 6.5: Comparison of the mean and the standard error of the residence time between four experimental systems

Next, we verified the *Prediction 2*. Figure 6.5 shows the mean and the standard error of the residence time for each system. A 2×2 two-way repeated-measure analysis of variance (ANOVA) was conducted ($N_h = 208.96$). A significant main effect in *interruptible* was revealed ($F(1, 856) = 39.22, p < .01, \eta^2 = .044$). Thus, we confirmed that the motion interruption makes HRI a more durable form of communication.

6.3 Human Impression

In previous Section 6.2, we confirmed the motion interruptivity makes communication more durable and the motion parameterization makes people easier to approach the Actroid. In this section, we evaluate the effectiveness of the motion parameterization after approaching the Actroid (*i.e.,* during communication).

6.3.1 Hypothesis and Prediction

The proposed motion parameterization makes people easier to approach the Actroid. We believe this result is caused by people perceived more positive impression of the Actroid. Thus, we made the following prediction.

· *Prediction*
  By the motion parameterization, people will perceive more positive impression, that is, better impression to the Actroid.

6.3.2 Conditions

In this section, we controlled only one condition, the motion parameterization. The interruptivity was always enabled, because we have already confirmed the interruptivity makes more responsive and durable communication. We assumed the result means people perceived better impression of the Actroid. Thus, we measured human impressions of the Actroid with two systems **I+NP** and **I+P** as mentioned in Section 6.2.2.

6.3.3 Method

Participants

Experiments were conducted with 42 subjects of various age groups (20 subjects for the **I+NP** and 22 subjects for the **I+P**). All subjects were visitors who attended an open house day of our university and voluntarily answered a questionnaire at the end of the communication with the Actroid. One of the systems was changed to another system every one hour. We let subjects interact freely, without suggesting anything, such as sitting position or timing during the communication process.

Settings

Subjects sit down on one of three seats placed in front of the Actroid, then communicate with each other for a few minutes. Figure 6.6 shows the snapshots of the communications with the condition **I+P**, when the speaker sat on (a) right-side seat, (b) center seat, and (c) left-side seat. As shown in Figure 6.6, the head and body of the Actroid faced the speaker's direction by motion parameterization. The other settings are the same as described in Section 6.2.3.

6.3.4 Measurement

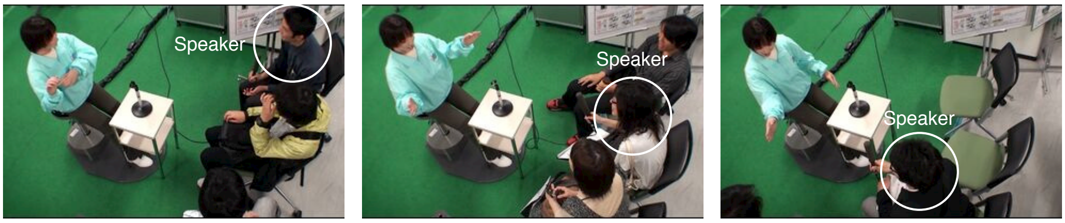
We used the Semantic Differential (SD) method [79] for the evaluation of human impressions of the android. After the communication with the Actroid was finished, each subject answered a questionnaire with 28 antonymous adjective pairs on a Likert scale from one to seven points (*i.e.,* one: the worst, seven: the best), as a SD profile. Each adjective pair is shown in Table 6.3. The higher score represents a positive impression, that is, a better impression.

6.3.5 Results

The results of the mean and the standard error of the SD profiles are shown in Figure 6.7. This figure shows that almost all scores in **I+P** are higher. Through detailed analysis we found that the parameterization increases not only the speaker's score, but also his/her neighbor's score.

Next, a factor analysis was conducted (eigenvalue ≥ 1, cumulative variance $\leq 50\%$, factor loading ≥ 0.5). Figure 6.8 shows four factor scores by *Bartlett* method [80] and *Promax* rotation [81]. Each factor contains some adjectives as shown in Table 6.4. Note that we name each factor based on its adjectives.

A student t-test revealed that *activity* ($p < .01$), *sophistication* ($p < .01$), *speediness* ($p < .01$), and *friendliness* ($p < .1$) in **I+P** are significantly higher than ones in



(a) Right-side (b) Center (c) Left-side

Figure 6.6: Snapshots of HRI experiments with the system **I+P** where a speaker is located (a) right-side, (b) center and (c) left-side of chairs

Table 6.3: 28 antonymous adjective pairs (left-side: positive, right-side: negative) described in the questionnaire

positive	negative	positive	negative
good	bad	sensitive	insensitive
kind	afraid	fulfilling	empty
cute	hateful	bright	dark
fun	boring	active	passive
warm	cold	familiar	unfamiliar
approachable	unapproachable	fast	slow
humanly	mechanical	quick	dull
cheerful	awful	interesting	uninteresting
friendly	unfriendly	considerate	selfish
likable	dislikable	complicated	simple
positive	negative	safe	dangerous
affable	disgusting	comprehensive	incomprehensive
wise	stupid	intense	mild
flashy	plain	strong	weak

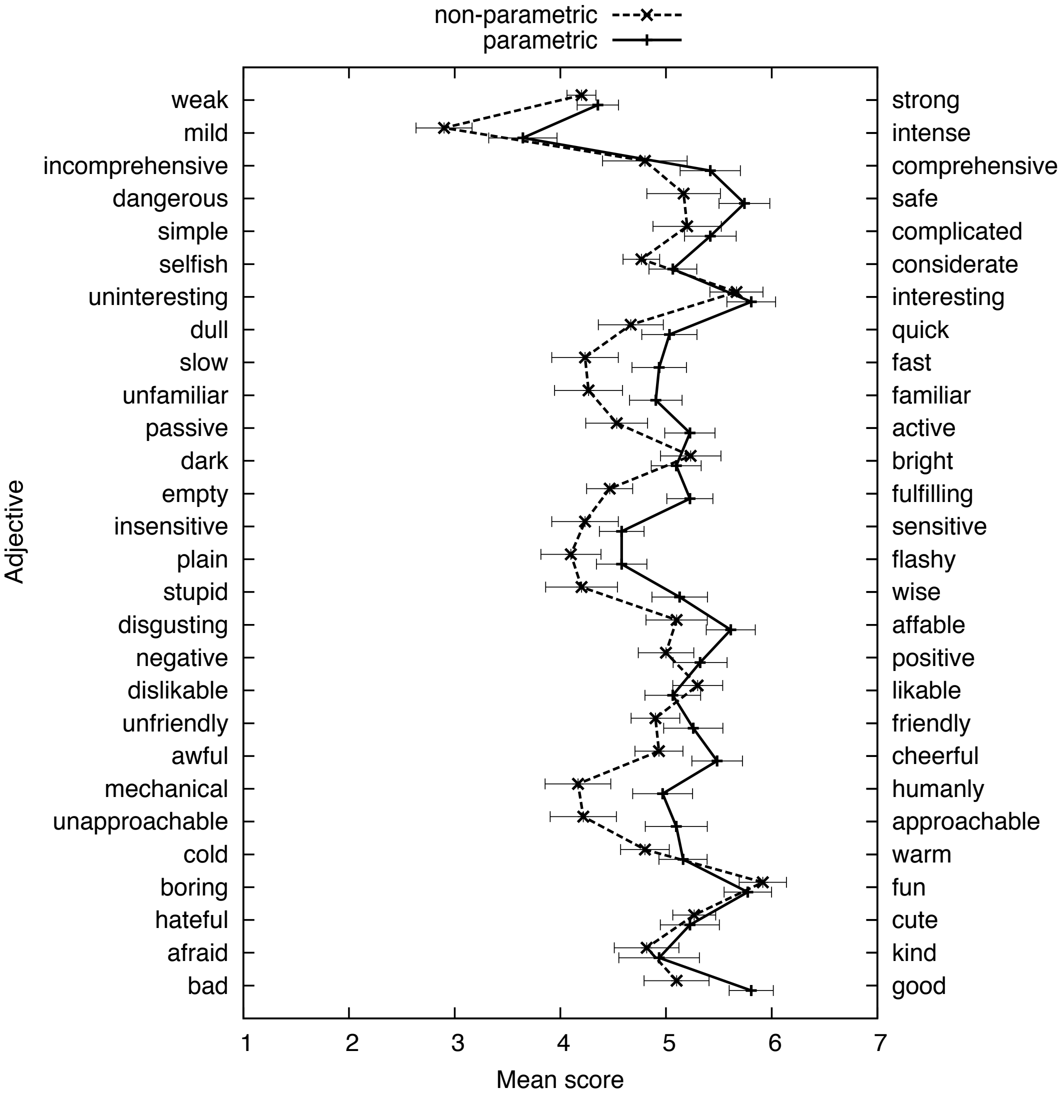


Figure 6.7: The mean and the standard error of the SD profiles of **I+NP** and **I+P** scores. We used a Likert scale from one to seven points (*i.e.,* one: the worst, seven: the best) for the questionnaire.

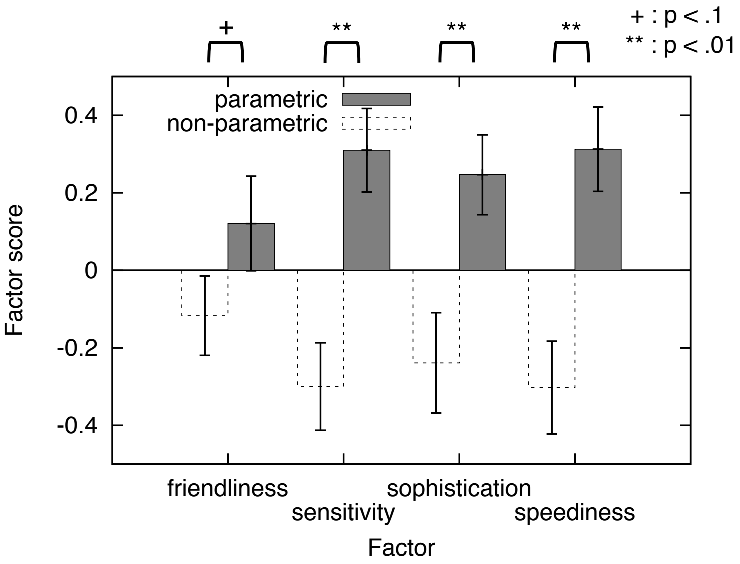


Figure 6.8: The mean scores and the standard errors of the four factors *friendliness*, *activity*, *sophistication*, and *speediness* by analyzing the SD profiles shown in Figure 6.7. The higher score represents a positive impression

Table 6.4: Four factors given by the factor analysis and their containing adjectives (positive ones only)

factor	adjectives
friendliness	cute, like, friendly, cheerful, positive, kind, fun
activity	active, fulfilling, sensitive, considerate
sophistication	comprehensive, approachable, wise
speediness	flashy, quick, fast, warm

I+NP. Therefore, our hypothesis which the motion parameterization makes more positive impression was supported. This result might indicate that the subjects implicitly perceived the Actroid as being wiser and moving quicker by the motion parameterization. Therefore, we concluded that body gesture planning is one of the most useful functions for a more human-like HRI system.

Chapter 7

Conclusion

7.1 Construction of Reconfigurable Motion Database

In this thesis, we focused on online body gesture planning for android based on a proposed *Reconfigurable Motion Database* (RMDB) in human-robot interaction (HRI) and human-robot symbiosis (HRS). The RMDB is the integration of data-driven methods and online planning methods. Hereby, user-intended interruption can be allowed in our HRI system while keeping features of original human-like motion in database. In addition, the RMDB can adjust gestures based on speaker and/or object location by motion parameterization and synthesis. Given a human-like motion sequence, which is easily obtained from a motion capture system, RMDB classifies similar motion sequences and memorizes them as a parametric gesture. The classification method called a *Bag-of-motion-features* can retrieve motions based on a semantic similarity. This method uses a wavelet as a local temporal feature of a motion. This idea is inspired by an image processing technique which expresses a local spatial information of an image in frequency domain. Motion is finally classified by comparing the *Bag-of-motion-features* which is also inspired by an image retrieval method, the *Bag-of-features*.

We conducted experiments using one-hour motions of 15 subjects to construct the motion database and evaluate its performance of the proposed method with several conventional methods. First, through the results of the accuracy of a pointing gesture and idle gesture, we confirmed that the mean errors of a location or angle parameters decrease monotonically in proportion to the number of sampling gestures. As a result, we had a common guidance to design a new parametric motion. Next, the number of correctly classified motions by the proposed method was larger than the others for all nine query gestures. The computational time of the motion database construction was also much shorter than the others. Finally, we also confirmed that the proposed method can interrupt motion to motion anytime without loss of key pose information and sharp velocity changes associated with self-collision. It seems to be maintained their motion features.

7.2 Application to Human-Robot Interaction

We developed a novel interaction system on an android *Actroid-SIT* which the RMDB is embedded for body gesture. In addition, our system can generate facial expression with five emotions, and includes gaze movement based on the knowledge of social animals. The system architecture is inspired by the *Episode Rule Selector*, and its components are connected via the *Key-Value Store*. The *Key-Value Store* has ACID (Atomicity, Consistency, Isolation, and Durability) properties. Therefore, we solve the issue on synchronization among components.

Experimental results revealed the effectiveness of the RMDB method for the improvement of human impressions. In first experiments, over 60 subjects attended the experiment to evaluate the effectiveness of proposed method, and we confirmed the feasibility of smooth communication, especially for children and seniors. Our system is considered about interruption on the way in HRI. Therefore, a robot can reply rapidly against human query.

To compare a speaker ratio (*i.e.,* the ratio of the number of people who start speaking to the Actroid, to the number of people who approach the Actroid) and the residence time of communication with or without the motion interruption and parameterization that we proposed, we conducted multi-party HRI experiments for 1,662 subjects in total. With our HRI system the speaker ratio was over 60%, though that of conventional systems was less than 50%. The residence time of communication was longer in our HRI system. Thus, interruptivity makes communication in HRI more durable and responsive.

By analyzing human impressions of the Actroid, we proved that motion parameterization contributes to the Actroid being wiser and more comprehensive. In these experiments, the Actroid generates appropriate gestures using the parameterization of RMDB, and can face in the speaker's direction. As a result of the SD method and factor analysis, a 1% level of significant differences between parametric and non-parametric gestures exist in activity, sophistication, and speediness factors. We found that the way of communication is dissimilar among age, gender, and character.

7.3 Future Work

We are now using only upper-body motion for body gestures. However, it is important to expand the proposed method into whole-body gestures including facial expression, because humanoids can control their whole-body motion and androids can control their facial expression. In addition, our method is original for motion retrieval and classification, but we hope that the method can be applied to online motion recognition.

Furthermore, we need to make the system more advanced by adding and/or upgrading its components. For example, we would like to implement a speech recognition component for multiple voices. If the component could be successful to work, we would easily evaluate the effectiveness of our HRI system with multiple people. We must also consider the automatic creation of episode rules by through a human history of tele-operation based on a *Wizard of Oz* method [82].

Acknowledgements

First of all, I would like to thank my professor and thesis adviser, Professor Tsukasa Ogasawara. He has welcomed me in the Robotics Laboratory and helped me in fulfilling my dream, that is, developing robotic systems. His unassuming ways of managing my research has helped me a lot in growing as a researcher and as a person. With his guidance generosity, I was able to accomplish this thesis. Without him, this thesis would not be a success.

A special thanks to my other thesis advisers, Associate Professor Jun Takamatsu and Assistant Professor Kentaro Takemura. They pushed me to greater heights and gave me much needed advice on how to go about my research. Their forthright and discerning guidance steered me toward the right direction and helped me clear the path towards the completion of this thesis.

Besides my advisers, I would also like to thank the rest of my thesis committee: Professor Kiyohiro Shikano. Thank you very much for reviewing my thesis and the invaluable guidance.

To Associate Professor Akihiko Yamaguchi and the Behavior Group members of the Robotics Laboratory, especially Emarc Magtanong, Gustavo Garcia, Satoki Tsuichihara. They have been a constant guide in my journey through this research and taught me different ways of looking at my work and approaching the problems I encountered.

In addition, a special thanks to Professor Tamim Asfour of Karlsruhe Institute of Technology, Germany and his lab-mates. When I was here as a visiting scholar for joining on a research group of a robot, *ARMAR* from September to October 2012, they were always kind to me and actively assisted me for the research. And also much appreciation goes to Dr. Takayuki Kanda of ATR Intelligent Robotics & Communication Laboratories. His heartily review and advice on my journal paper helped this thesis how to properly evaluate the subject experiments.

I would also like to thank all the members of the Robotics Laboratory, past and present, many of whom have become like a family to me. Special mention to Assistant Professor Atsutoshi Ikeda, Associate Professor Yuichi Kurita (currently in Hiroshima University), Junichi Ido, Tsuyoshi Suenaga, Albert Causo, Ding Ming, Yumiko Suzuki, Masato Kawamura, Ato Araki, and Yoichiro Yamagi.

Let me also thank the administrative staffs in the lab who have been patient with me and helped me with the bureaucratic side of being a student: Miyuki Yamaguchi, Michiko Owaki, and Megumi Kanaoka.

I would like to thank all my friends here for joining me in my life travels all throughout. In particular, my colleagues who had been always together in my master's course of NAIST, Yuji Kohashi, Yoshiyuki Takeuchi, Takeshi Tamaki, Yuya Hayama, Kotaro Hirasawa, Shun Hirose, Noriyuki Matsunaga, and Kenta Mukai. I send you all my endless thanks.

And to my loving and caring parents and family. For being my source of strength and hope, for always being there, for reminding me of the things I love and helping me remember why I do what I do. Thanks!

List of Publications

Refereed Journal Papers

1. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Gesture-Centric Android System for Multi-Party Human-Robot Interaction," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 133–151, February 2013.

2. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Responsive Gesture Planning based on Integration of Motion Database and Online Planning," *Journal of the Robotics Society of Japan*, vol. 30, no. 9, pp. 899–906, November 2012 (in Japanese).

Refereed International Conference Proceedings Papers

1. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Gesture-Centric Android System for Multi-Party Human-Robot Interaction," *in Proceedings of the 8th ACM/IEEE International Conference on Human Robot Interaction*, Tokyo, Japan, March 2013 (on journal session).

2. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Body Gesture Classification based on Bag-of-features in Frequency Domain of Motion," *in Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 386–391, Paris, France, September 2012.

3. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Planning Body Gesture of Android for Multi-person Human-Robot Interaction," *in Proceedings of the 2012 IEEE International Conference of Robotics and Automation*, pp. 3897–3902, St. Paul, USA, May 2012.

4. **Yutaka Kondo**, Masato Kawamura, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Gaze Motion Planning for Android Robot," *in Proceedings of the 6th ACM/IEEE International Conference on Human Robot Interaction*, pp. 171–172. Lausanne, Switzerland, March 2011.

5. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Multi-person Human-Robot Interaction System for Android Robot," *in Proceedings of the 2010 IEEE/SICE International Symposium on System Integration*, pp. 176–181, Sendai, Japan, December 2010.

6. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Smooth Human-Robot Interaction by Interruptible Gesture Planning," *in Proceedings of the 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 213–218, Montreal, Canada, July 2010.

Refereed Japanese Conference Proceedings Papers

1. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Conceptual Motion Retrieval based on Bag-of-features in Frequency Domain of Motion," *in Proceedings of the 17th Annual Conference of the Robotics Symposia*, pp. 541–547, March 2012.

Japanese Conference Proceedings Papers

1. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Similar Motion Classification based on Continuous Wavelet Transform for Construction of Gesture Database," *the 29th Annual Conference of the Robotics Society of Japan*, 1F3-4, September 2011.

2. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Body Gesture Planning under Multi-person Human-Robot Symbiosis," *the 28th Annual Conference of the Robotics Society of Japan*, 3B2-2, September 2010.

3. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Multiple Interruption-aware HRI System for Smooth Interaction," *the Robotics and Mechatronics Conference 2010*, 1P1-D03, June 2010.

4. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Online-planning of Flexible Interactive Motion based on Reconfigurable Database," *the 27th Annual Conference of the Robotics Society of Japan*, 1L3-05, June 2009.

5. **Yutaka Kondo**, Junichi Ido, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "RT-Components for Multithreaded Motion Planning," *the Robotics and Mechatronics Conference 2009*, 2A2-C14, May 2009.

Patents

1. **Yutaka Kondo**, Kentaro Takemura, Jun Takamatsu and Tsukasa Ogasawara, "Path Planner and Control Device of Robotic System," *Japanese Patent P110002346*, September 2009 (in Japanese).

Awards

1. **Yutaka Kondo**, "Best Student Award," *Nara Institute of Science and Technology*, March 2013.

References

[1] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, “Robovie: An interactive humanoid robot,” *Journal of Industrial Robot*, vol. 28, no. 6, pp. 498–503, 2001.

[2] Internal Affairs and Communications Ministry in Japan, *To Realize Next IT Network Robot made in Japan*. Conference Report of Network Robot Technology, 2003.

[3] J. Fast, *Body Language*. Simon & Schuster Adult Publishing Group, 1970.

[4] K. Shinozawa, F. Naya, J. Yamato, and K. Kogure, “Differences in effect of robot and screen agent recommendations on human decision-making,” *International Journal of Human-Computer Studies*, vol. 162, no. 2, pp. 267–279, 2005.

[5] M. Mori, “Bukimi no tani (the uncanny valley),” *Energy*, vol. 7, no. 4, pp. 33–35, 1970.

[6] G. White, L. McKay, and F. Pollick, “Motion and the uncanny valley,” *Journal of Vision*, vol. 7, no. 9, p. 477, 2007.

[7] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. Kumar, K. D. Meng, and G. W. Kit, “Davinci: A cloud computing framework for service robots,” in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3084–3089, 2010.

[8] G. Hu, W. P. Tay, and Y. Wen, “Cloud robotics: Architecture, challenges and applications,” *Journal of IEEE Network*, no. 3, pp. 21–28, 2012.

[9] K. Kamei, S. Nishio, N. Hagita, and M. Sato, “Cloud networked robotics,” *Journal of IEEE Network*, no. 3, pp. 28–34, 2012.

[10] T. Flash and N. Hogan, “The coordination of arm movements: An experimentally confirmed mathematical model,” *Journal of Neuro Science*, vol. 5, no. 7, pp. 1688–1703, 1985.

[11] T. Chikaraishi, T. Minato, and H. Ishiguro, “Development of an android system integrated with sensor networks,” *Journal of Information Processing Society of Japan*, vol. 49, no. 12, pp. 3821–3834, 2008.

[12] L. Kovar, M. Gleicher, and F. Pighin, “Motion graphs,” in *Proceedings of ACM SIGGRAPH*, pp. 473–482, 2002.

[13] T. Inamura, Y. Nakamura, and I. Toshima, “Embodied symbol emergence based on mimesis theory,” *International Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[14] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Marcov Models for Speech Recognition*. Edinburgh University Press, 1990.

[15] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[16] S. Lazebnik, C. Schmid, and J. Ponce, “Object recognition from local scale invariant features,” in *Proceedings of IEEE International Conference on Computer Vision*, pp. 1150–1157, 1999.

[17] L. Kovar and M. Gleicher, “Automated extraction and parameterization of motions in large data sets,” *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 559–568, 2004.

[18] L. E. Kavraki, P. Svestka, J. Latombe, and M.H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transaction on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[19] K. Nagao and A. Takeuchi, “Social interaction: Multimodal conversation with social agents,” in *Proceedings of National Conference on Artificial Intelligence*, pp. 22–28, 1994.

[20] Y. Matsusaka, T. Tojo, and T. Kobayashi, “Conversation robot participating in group conversation,” *IEICE Transaction on Information and Systems*, vol. E86-D, no. 1, pp. 26–36, 2003.

[21] D. Sakamoto, T. Kanda, T. Ono, H. Ishiguro, and N. Hagita, “Android as a telecommunication medium with a human-like presence,” in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*, pp. 193–200, 2007.

[22] A. T. Bahill, D. Adler, and L. Stark, “Most naturally occurring human saccades have magnitudes of 15 deg or less,” *Invest Ophthalmol & Visual Science*, vol. 14, pp. 468–469, 1975.

[23] E. Bizzi, R. E. Kalil, and V. Tagliasco, “Eye-head coordination in monkeys,” *Science*, vol. 173, pp. 452–454, 1971.

[24] M. Yamada, “Analysis of head and eye co-ordination when viewing targets on a two-dimensional plane,” *Technical Report of Institute of Electronics, Information and Communication Engineers*, vol. 75, no. 5, pp. 971–981, 1992.

[25] Y. Uno, M. Kawato, and R. Suzuki, “Formation and control of optical trajectory in human multi-joint arm movement – minimum torque-change model,” *Biological Cybernetics*, vol. 61, no. 2, pp. 89–101, 1989.

[26] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, “Adaptive response of a gene network to environmental changes by fitness-induced attractor selection,” *Journal of PLos ONE*, vol. 1, no. e49, pp. 1–10, 2006.

[27] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 473–479, 1999.

[28] J. J. Kuffner and S. M. LaValle, “Rrt-connect: An efficient approach to single-query path planning,” in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 995–1001, 2000.

[29] B. Scassellati, “Investigating models of social development using a humanoid robot,” in *Proceedings of International Joint Conference on Neural Networks*, pp. 2704–2709, 2003.

[30] H. Ogawa and T. Watanabe, “Interrobot speech-driven embodied interaction robot,” *Journal of Advanced Robotics*, vol. 15, no. 3, pp. 371–377, 2001.

[31] S. Masuko and J. Hoshino, "Generating head-eye movement for virtual actor," *Technical Report of Institute of Electronics, Information and Communication Engineers*, vol. 88, no. 3, pp. 585–595, 2005.

[32] A. Fukayama, P. VicentBao, and T. Ohno, "Analysis of user's gaze for usability assessment of anthropomorphic agents," *Technical Report of Institute of Electronics, Information and Communication Engineers*, vol. 103, no. 743, pp. 53–58, 2004.

[33] K. Brügmann, H. Dohrn, and H. Prendinger, "Phase-based gesture motion parametrization and transitions for conversational agents with mpml3d," in *Proceedings of 2nd International Conference on INtelligent TEchnologies for interactive enterTAINment*, no. 10, pp. 1–6, 2008.

[34] C. Rose, B. Bodenheimer, and M. F. Cohen, "Verbs and adverbs: multidimensional motion interpolation," *Journal of IEEE Computer Graphics and Applications*, no. 5, pp. 32–40, 1998.

[35] J. Liu and M. Shah, "Learning human actions via information maximization," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[36] M. Raptis, K. Wnuk, and S. Soatto, "Flexible dictionaries for action classification," in *IEEE International Workshop on Machine Learning for Vision-based Motion Analysis*, 2008.

[37] M. Raptis, D. Kirovski, and H. Hoppes, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 147–156, 2011.

[38] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proceedings of ACM International Conference on Computer Vision*, pp. 1036–1043, 2011.

[39] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *Proceedings of ACM International Conference on Computer Vision*, pp. 1593–1600, 2009.

[40] C. Schüldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local svm approach,” in *Proceedings of ACM International Conference on Pattern Recognition*, pp. 32–36, 2004.

[41] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[42] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis,” in *Proceedings of Eurospeech*, pp. 2347–2350, 1999.

[43] Z. H. Ling, Y. J. Wu, Y. P. Wang, L. Qin, and R. H. Wang, “Ustc system for blizzard challenge 2006 an improved hmm-based speech synthesis method,” in *Blizzard Challenge Workshop*, 2006.

[44] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, “Details of nitech hmm-based speech synthesis system for the blizzard challenge 2005,” *IEICE Transaction on Information and Systems*, no. 1, pp. 325–333, 2007.

[45] A. W. Black, H. Zen, and K. Tokuda, “Statistical parametric speech synthesis,” in *Proceedings of ICASSP*, pp. 1229–1232, 2007.

[46] J. Yu, M. Zhang, J. Tao, and X. Wang, “A novel hmm-based tts system using both continuous hmms and discrete hmms,” in *Proceedings of ICASSP*, pp. 709–712, 2007.

[47] T. Inamura, “Recognition, teaching and generation of human’s motion by hmm,” *Journal of Robotics Society of Japan*, vol. 29, no. 5, pp. 419–422, 2011.

[48] J. Ido, Y. Matsumoto, T. Ogasawara, and R. Nishimura, “Humanoid with interaction ability using vision and speech information,” in *Proceedings of IEEE International Conference on Robots and Systems*, pp. 1316–1321, 2006.

[49] D. Lee, C. Ott, and Y. Nakamura, “Mimetic communication with impedance control for physical human-robot interaction,” in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1535–1542, 2009.

[50] S. Haddadin, A. Albu-Schäffer, A. D. Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3356–3363, 2008.

[51] Z. Miyashita, T. Kanda, M. Shiomi, H. Ishiguro, and N. Hagita, "A robot in a shopping mall that affectively guide customers," *Journal of Robotics Society of Japan*, vol. 26, no. 7, pp. 103–114, 2008.

[52] E. A. Sisbot, R. Alami, T. Simeon, K. Dautenhahn, M.Walters, S. Woods, K. L. Koay, and C. Nehaniv, "Navigation in the presence of humans," in *Proceedings of IEEE/RAS International Conference on Humanoid Robots*, pp. 181–188, 2005.

[53] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "An affective guide robot in a shopping mall," in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*, pp. 173–180, 2009.

[54] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita, "Developing a model of robot behavior to identify and appropriately respond to implicit attention-shifting," in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*, pp. 133–140, 2009.

[55] R. E. Kraut, S. R. Fussell, and J. Siegel, "Visual information as a conversational resource in collaborative physical tasks," *Human-Computer Interaction*, vol. 18, no. 1, pp. 13–49, 2003.

[56] O. Morikawa and T. Maesako, "Hypermirror: Toward pleasant-to-use video mediated communication system," in *Proceedings of Conference on Computer supported cooperative work*, pp. 149–158, 1998.

[57] Y. Matsusaka, S.Fujie, and T. Kobayashi, "Framework of communication activation robot participating in multiparty conversation," in *Proceedings of AAAI Fall Symposium, Dialog with Robots*, pp. 68–73, 2010.

[58] H. Nakanishi, S. Nakazawa, T. Ishida, K. Takanashi, and K. Isbister, "Can software agents influence human relations?: Balance theory in agent-mediated communities," in *Proceedings of ACM International Joint Conference on Autonomous agents and multiagent systems*, pp. 717–724, 2003.

[59] D. Sakamoto and T. Ono, "Sociality of robots: Do robots construct or collapse human relations?" in *Proceedings of ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pp. 355–356, 2006.

[60] T. Minato, M. Shimada, H. Ishiguro, and S. Itakura, "Development of an android robot for studying human-robot interaction," in *Proceedings of International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp. 424–434, 2004.

[61] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[62] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of European Conference on Computer Vision*, pp. 430–443, 2006.

[63] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision Image Understanding*, vol. 110, no. 3, pp. 246–259, 2008.

[64] P. Goupillauda, A. Grossmanna, and J. Morlet, "Cycle-octave and related transforms in seismic signal analysis," *Seismic Signal Analysis and Discrimination III*, vol. 23, no. 1, pp. 85–102, 1984.

[65] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.

[66] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[67] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1997.

[68] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *AAAI Workshop on Knowledge Discovery in Databases*, pp. 359–370, 1994.

[69] A. Nakazawa, S. Nakaoka, and K. Ikeuchi, "Synthesize stylistic human motion from examples," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3899–3904, 2003.

[70] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and L. E. Kavraki, "Sampling-based roadmap of trees for parallel motion planning," *IEEE Transaction on Robotics*, vol. 21, no. 4, pp. 597–609, 2005.

[71] D. Demirdjian Y. Song and R. Davis, "Tracking body and hands for gesture recognition: Natops aircraft handling signals database," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 500–506, 2011.

[72] M. Isard and A. Blake, "Condensation-conditional density propa- gation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[73] S. Chen, Y.Tian, Q. Liu, and D. N. Metaxas, "Recognizing expressions from face and body gesture by temporal normalized motion and appearance features," *Image and Vision Computing*, pp. 1–11, 2012.

[74] E. F. Codd, "A relational model of data for large shared data banks," *Communications of ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[75] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[76] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W. Yoon, "Rt-middleware: Distributed component middleware for rt (robot technology)," in *Proceedings of IEEE International Conference on Robots and Systems*, pp. 3555–3560, 2005.

[77] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[78] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 511–518, 2001.

[79] C. E. Osgood and P. Tannenbaum, *The Measurement of Meaning*. University of Illinois Press, 1957.

[80] M. S. Bartlett, "The statistical conception of mental factors," *British Journal of Psychology*, vol. 28, no. 1, pp. 97–104, 1937.

[81] A. E. Hendrickson and P. O. White, "Promax: A quick method for rotation to oblique simple structure," *British Journal of Psychology*, vol. 17, no. 1, pp. 65–70, 1964.

[82] N. M. Fraser and G. N. Gilbert, "Simulating speech systems," *Journal of Computer Speech and Language*, vol. 5, no. 1, p. 8199, 1991.

[83] M. Mizuguchi, J. Buchanan, and T. Calvert, "Data driven motion transitions for interactive games," *Eurographics 2001 Short Presentations*, 2001.

Appendix

A. Motion Retargeting Algorithm

Captured motions need to transform from a human to a robot coordinates. In this research, it is required to register these two coordinates, because there are lots of subjects and they move freely. The invariant transform is also utilized to calibrate the motions. Therefore, we define a dissimilarity between human and robot appearances as Equation (A.1), which is defined as the distance between all of human's retroreflective marker $\mathbf{p}'_i\big(= (x'_i, y'_i, z'_i)\big)$ and corresponding robot's marker $\mathbf{p}_i$. We generate robot motion sequences by minimizing the equation.

$$\arg\min_{\theta, x_o, y_o} \sum_i w_i \|\mathbf{p}_i - \mathbf{T}_{\theta, x_o, y_o} \mathbf{p}'_i\|^2. \tag{A.1}$$

The coordinate system is shown in Figure 3.1 (b). $\mathbf{T}_{\theta,x_o,y_o}$ indicates top-view rigid two dimensional transformation matrix. x_o and y_o are translation, θ is an angle of rotation about z axis, and the weight coefficient w_i is chosen empirically to assign more important markers. This optimization has a closed-form solution [83]:

$$\theta = \arctan \frac{\sum_i w_i (x_i y'_i - x'_i y_i) - \frac{1}{\sum_i w_i} (\bar{x}_i \bar{y}'_i - \bar{x}'_i \bar{y}_i)}{\sum_i w_i (x_i x'_i + y_i y'_i) - \frac{1}{\sum_i w_i} (\bar{x}_i \bar{x}'_i + \bar{y}_i \bar{y}'_i)}, \tag{A.2}$$

$$x_o = \frac{1}{\sum_i w_i} (\bar{x} - \bar{x}' \cos\theta - \bar{y}' \sin\theta), \tag{A.3}$$

$$y_o = \frac{1}{\sum_i w_i} (\bar{y} + \bar{x}' \sin\theta - \bar{y}' \cos\theta), \tag{A.4}$$

where $\bar{x} = \sum_i w_i x_i$ and the other terms with bar are defined similarly. Figure A.1 shows an example to transform a pointing gesture. The blue and red points in the middle row of Figure A.1, indicate the positions of human's retroreflective markers $\mathbf{p}'$ and robot's markers $\mathbf{p}$, respectively. We eliminated motions whose dissimilarity calculated by Equation (A.1) is more than a threshold.

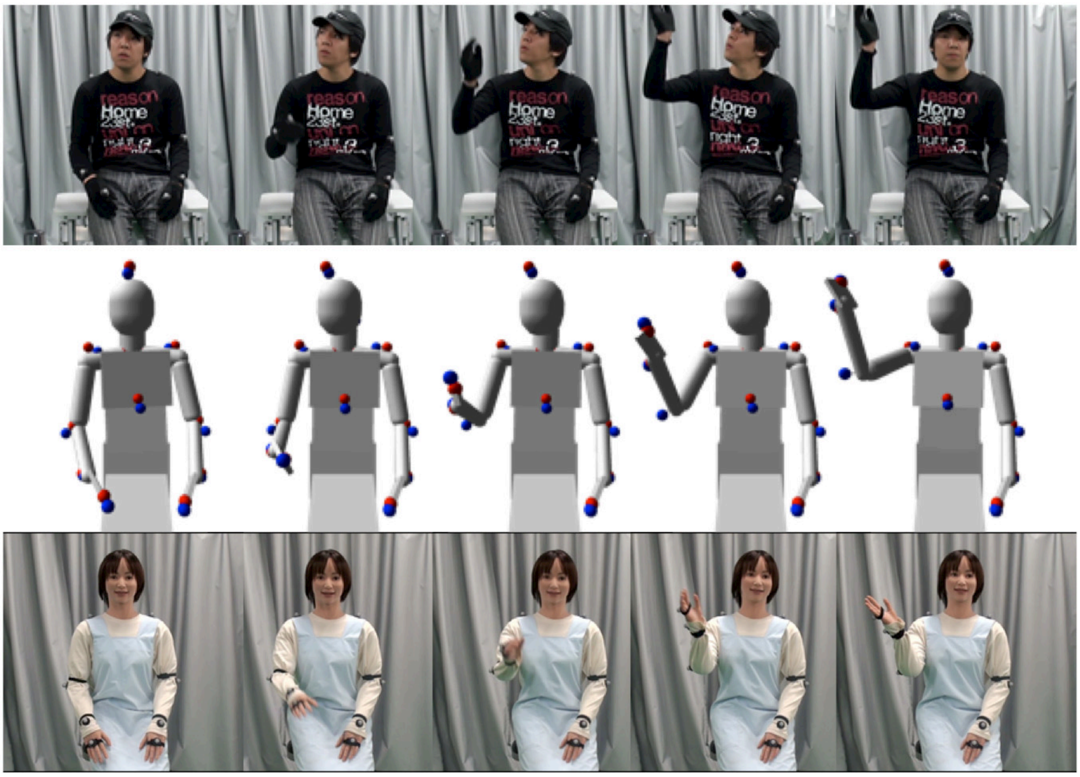


Figure A.1: The results of conversions to appropriate Actroid-SIT configuration from captured pointing sequence

B. Gaze Movement Planning

Masuko *et al.* [31] proposed a method for gaze movement of a CG avatar. We applied this method for the Actroid's gaze movement.

B.1 Sharing Rate and Convergence

Gaze angle V_x indicates the angle between the front direction and the gaze direction as shown in Figure B.1. V_x is the sum of the head angle H_x between the head direction and the gaze direction E_x. The sharing rate $D_x = \frac{E_x}{V_x}$ indicates the ratio of eye movement to head movement in gaze motion. Although the sharing rate is altered by ages and/or sexes, in this paper, we employ the sharing rate defined in Equation (B.1).

$$D_x = \begin{cases} 1 & |V_x| \leq \frac{\pi}{12} \\ \frac{\frac{\pi}{6} - |V_x|}{\frac{\pi}{12}} & \frac{\pi}{12} \leq |V_x| \leq \frac{\pi}{6}. \\ 0 & \textit{otherwise} \end{cases} \tag{B.1}$$

Eye convergence occurs due to a gap in the direction from both eyes, when gazing at an object located at finite distance. Let (x, y, z) be an object position. Given that the direction of the object is E_x, the direction of left eye E_{xl}, and that of the right eye E_{xr}, $E_{xl} + E_{xr} = \frac{E_x}{2}$ can be satisfied. Then, we generate angles V_x and H_x considering the sharing rate and convergence by the following equations:

$$V_x = \tan^{-1}\left(\frac{x}{z}\right), \tag{B.2}$$

$$H_x = (1 - \alpha D_x)\, V_x, \tag{B.3}$$

where α $(0 \leq \alpha \leq 1)$ is for adjustment of the convergence ratio, which is empirically assigned. Next, given an object point $(\grave{x}, \grave{y}, \grave{z})$ calculated by (x, y, z) rotating H_x along head coordinates, we measure $(\grave{x} - \frac{l}{2}, \grave{y}, \grave{z} + d)$ as the length from the right eye to the object and $(\grave{x} + \frac{l}{2}, \grave{y}, \grave{z} + d)$ as the same for the left eye. We calculate angles E_{xl}, E_{xr} by Equations (B.4) and (B.5). Finally, we can plan the gaze motion as shown in Figure B.2. There are examples of angle V_x as $-\frac{\pi}{3}, -\frac{\pi}{6}, +\frac{\pi}{6}, +\frac{\pi}{3}$, respectively.

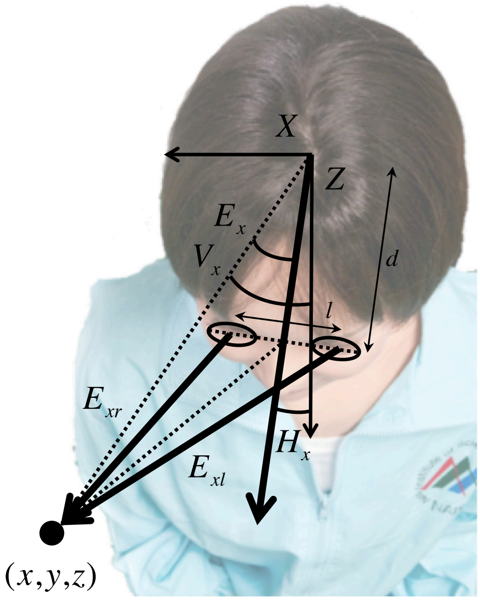


Figure B.1: The definition of the angles of the head and both eyes by considering the sharing rate and convergence

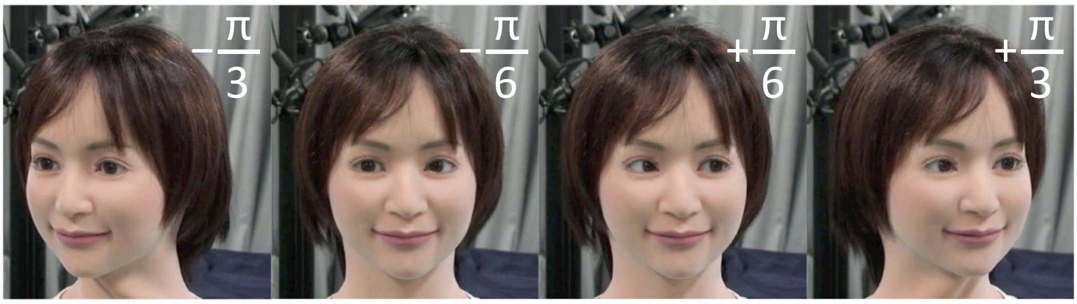


Figure B.2: Examples of gaze motions facing angle $V_x = -\frac{\pi}{3}, -\frac{\pi}{6}, +\frac{\pi}{6}, +\frac{\pi}{3}$, respectively

$$E_{xl} = \tan^{-1}\left(\frac{\grave{x}+\frac{l}{2}}{\grave{z}}\right), \tag{B.4}$$

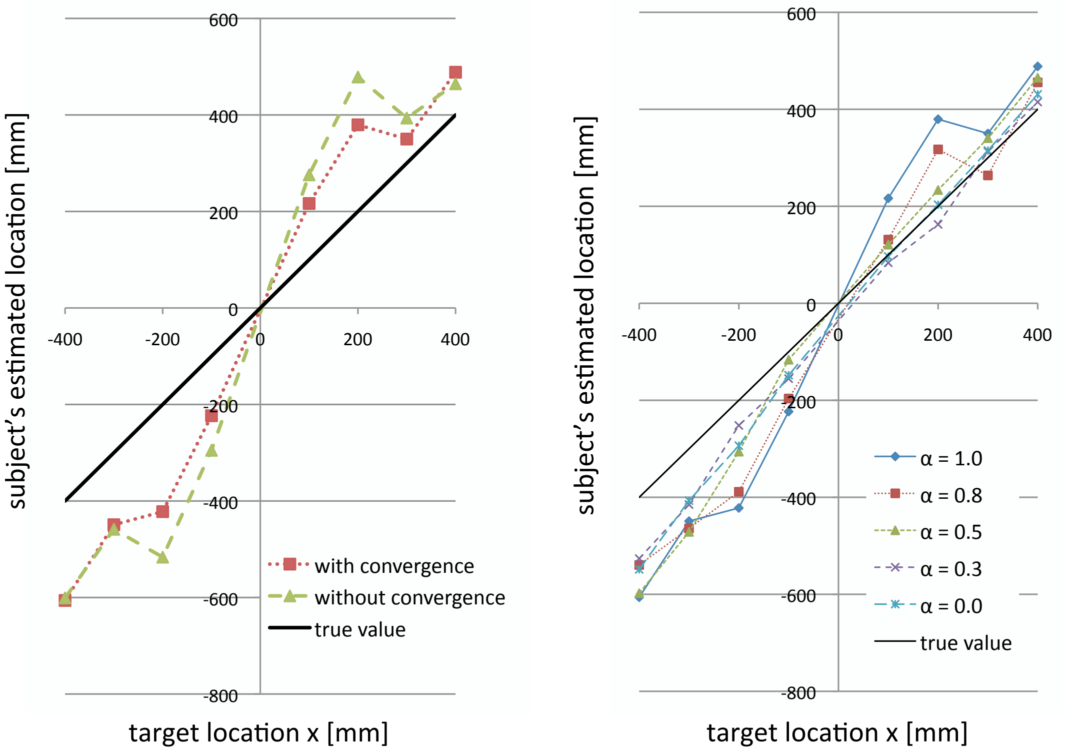
$$E_{xr} = \tan^{-1}\left(\frac{\grave{x}-\frac{l}{2}}{\grave{z}}\right). \tag{B.5}$$

B.2 Perception of Gaze

It is required to determine the α for the evaluation of the convergence and sharing rate. We conducted two comparison experiments. The Actroid first faces front for three seconds, then performs gaze motion toward $(x$, 500 [mm], 0 [mm]$)$ where $|x| \leq$ 400 [mm] at 100 [mm] intervals randomly for five seconds. The experiments were conducted for 11 subjects. Each subject stands 1800 [mm] from the Actroid, and marks perceived locations on the bar placed between the subject and Actroid, eight times in total.

Figure B.3 (a) and Figure B.3 (b) show the relation between target location x and subject's estimated location during the experiments of convergence and sharing rate, respectively. As shown in Figure B.3 (a), in the case of existence of convergence the estimation was closer to the true value than that in the case of absence of convergence. The data among $|x| \leq$ 200 [mm] has 5% level of significant difference by t-tests.

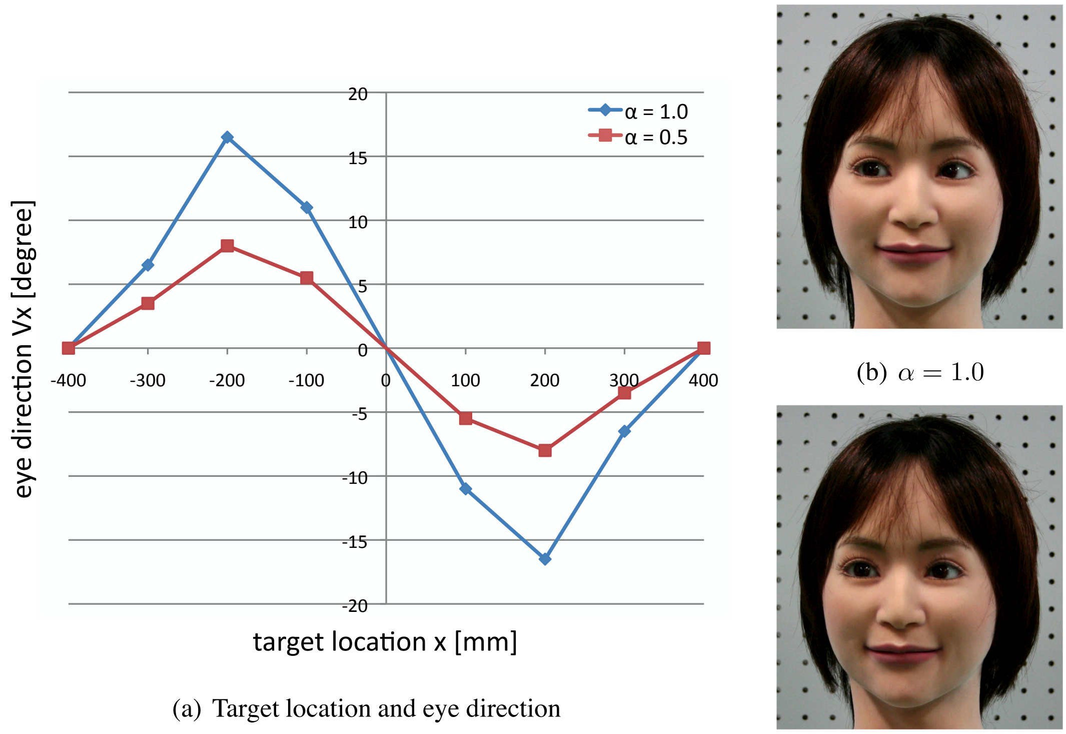
Figure B.3 (b) shows the effects of the sharing rate. The smaller α values (*i.e.,* the smaller eye movements), the closer to the true value. The data between $\alpha = 1.0$ and 0.5 when $|x| \leq$ 200 [mm], has 5% level of significant difference by t-tests. However, Bahill *et al.* [22] claimed that 86% of humans expresses gaze motion only by eye movement while their gaze direction is within $\frac{\pi}{12}$. Figure B.4 shows the relation between target location x and eye direction E_x, and the Actroid's face appearance when $\alpha = 1.0, 0.5$, respectively. In the result of $\alpha = 1.0$, the eye direction $E_x > \frac{\pi}{12}$ when $x = \pm$200 [mm]. The Actroid presents overestimation of gaze direction and the face poses tend to be more to the right, as you can see in Figure B.4. Because of that, we used $\alpha = 0.5$ in this paper.



(a) Convergence

(b) Sharing rate

Figure B.3: Comparison of the precision (a) with or without convergence, and (b) several α ratios.



(a) Target location and eye direction

(b) $\alpha = 1.0$

(c) $\alpha = 0.5$

Figure B.4: Relation between target location x and eye direction E_x: (a) the chart and the Actroid's appearance when $\alpha =$ (b) 1.0 and (c) 0.5

C. Design of Episode Rule

Our HRI system has around 200 episode rules. The episode rules are described using a script language, *Jython*. In this section, we describe how to design the episode rules using three examples. The examples indicate the response to the question "What's your name?" (Figure C.1), "My name is..." (Figure C.2), and "You are cute." (Figure C.3). The following three functions are defined in their scripts.

checkPrecondition(self, words, location)

This function has two arguments. The words contains a set of pairs which have a keyword and its score. The location contains the three dimensional location of a speaker. According to their sensory data and the history of interaction which the HRI system has, this function returns a value.

generateInteraction(self)

If a rule whose score of the function **checkPrecondition** was the highest, its other functions **generateInteraction** and **applyPosteffect** are called in order. The **generateInteraction** returns next interaction scenario which contains a type of gesture and its parameter, response sentences, and an emotion.

applyPosteffect(self)

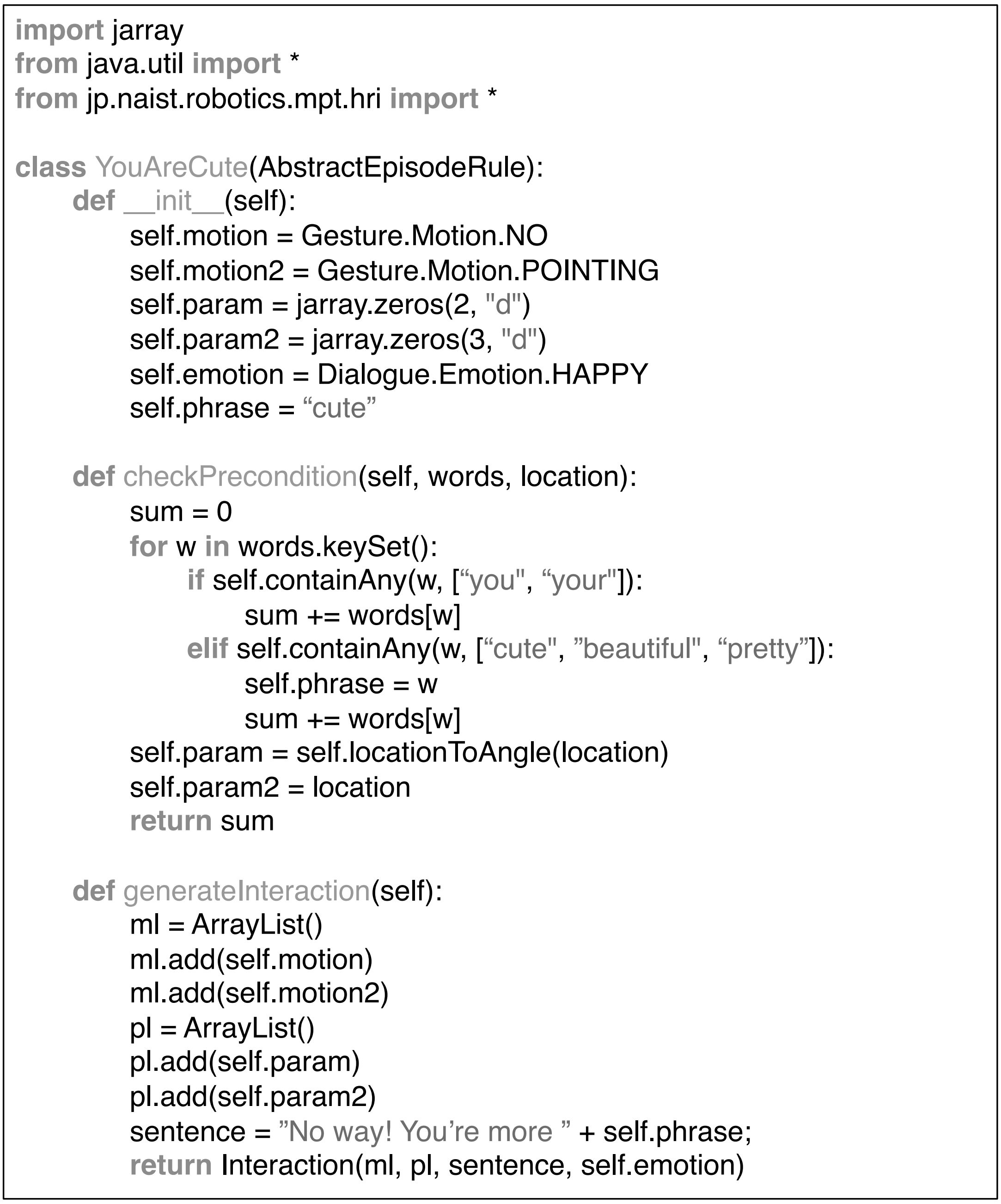
This function is called to set the posteffect of the selected rule, when a current interaction is finished. The posteffect is utilized for next continuous interaction.

```
import jarray
from java.util import *
from jp.naist.robotics.mpt.hri import *

class WhatYourName(AbstractEpisodeRule):
    def __init__(self):
        self.motion = Gesture.Motion.POINTING_MYSELF
        self.param = jarray.zeros(2, "d")
        self.emotion = Dialogue.Emotion.NORMAL

    def checkPrecondition(self, words, location):
        if SharedData.getBoolean("hri.alreadyAskedName"):
            return 0
        sum = 0
        for w in words.keySet():
        if self.containAny(w, [“you”, “name", “what", ”who"]):
            sum += words[w]
        self.param = self.locationToAngle(location)
        return sum

    def generateInteraction(self):
        sentence = “My name is " + SharedData.get(u"robot.name”) \
            + ”. What’s your name?”
        return Interaction(self.motion, self.param, sentence, self.emotion)

    def applyPosteffect(self):
        SharedData.setBoolean("hri.alreadyAskedName", True)
```

Figure C.1: The episode rule script for the question “What’s your name?”

```
import jarray
from java.util import *
from jp.naist.robotics.mpt.hri import *

class MyNameIs(AbstractEpisodeRule):
    def __init__(self):
        self.motion = Gesture.Motion.BOTH_SPREADING
        self.param = jarray.zeros(3, "d”)
        self.emotion = Dialogue.Emotion.NORMAL
        self.speaker = ””

    def checkPrecondition(self, words, location):
        if SharedData.getBoolean("hri.knowSpeakerName"):
            return 0
        sum = 0
        if SharedData.getBoolean("hri.alreadyAskedName"):
            sum += 100
        for w in words.keySet():
            if self.containAny(w, [”I“, ”my“, ”name"]):
                sum += words[w]
        self.speaker = self.primeKeyword(words)
        return sum

    def generateInteraction(self):
        sentence = “Your name is " + self.speaker + ”, isn’t it?”
        return Interaction(self.motion, self.param, sentence, self.emotion)

    def applyPosteffect(self):
        SharedData.setBoolean("hri.knowSpeakerName", True)
        SharedData.set("speaker.name", self.speaker)
```

Figure C.2: The episode rule script for the question “My name is ...”



```
import jarray
from java.util import *
from jp.naist.robotics.mpt.hri import *

class YouAreCute(AbstractEpisodeRule):
    def __init__(self):
        self.motion = Gesture.Motion.NO
        self.motion2 = Gesture.Motion.POINTING
        self.param = jarray.zeros(2, "d")
        self.param2 = jarray.zeros(3, "d")
        self.emotion = Dialogue.Emotion.HAPPY
        self.phrase = “cute”

    def checkPrecondition(self, words, location):
        sum = 0
        for w in words.keySet():
            if self.containAny(w, [“you", “your"]):
                sum += words[w]
            elif self.containAny(w, [“cute", ”beautiful", “pretty”]):
                self.phrase = w
                sum += words[w]
        self.param = self.locationToAngle(location)
        self.param2 = location
        return sum

    def generateInteraction(self):
        ml = ArrayList()
        ml.add(self.motion)
        ml.add(self.motion2)
        pl = ArrayList()
        pl.add(self.param)
        pl.add(self.param2)
        sentence = ”No way! You’re more ” + self.phrase;
        return Interaction(ml, pl, sentence, self.emotion)
```

Figure C.3: The episode rule script for the question “You are cute.”