

# 異種ロボット間でのジェスチャ情報の共有化

広瀬健志郎<sup>†</sup> 川島 英之<sup>††</sup> 佐竹 聡<sup>†††</sup> 今井 倫太<sup>††††</sup>

<sup>†</sup> 日本 IBM

<sup>††</sup> 筑波大学大学院システム情報工学研究科

<sup>†††</sup> 慶應義塾大学大学院開放環境科学専攻

<sup>††††</sup> 慶應義塾大学理工学部情報工学科

E-mail: <sup>†</sup>{hirose,satake}@ayu.ics.keio.ac.jp, <sup>††</sup>kawasima@cs.tsukuba.ac.jp, <sup>†††</sup>michita@ics.keio.ac.jp

あらまし 本研究の目的は、異種ロボット間でのジェスチャ情報を共有化することである。本研究では、ロボットの意味情報を共有するために、ジェスチャの本質的な特徴に注目しジェスチャを2種類に分類した。その2種類とは直示ジェスチャと軌道重視ジェスチャである。直示ジェスチャはどこかを指差すジェスチャであり、最後のポーズが本質的な特徴である。軌道重視ジェスチャは、ジェスチャの軌道自体に意味があり、軌道を平行移動させたり縮小した場合でも本質的な特徴を保存できる。それぞれの分類毎にジェスチャの変換アルゴリズムを作成したことで、ジェスチャの意味情報の共有が実現できた。ジェスチャ変換アルゴリズムを評価するために、2つの評価実験を行った。1つ目は評価シミュレーションを用いた定量的評価であり、2つ目はアンケートによる定性的評価である。両実験の結果、関節を消失したロボットに対して、ジェスチャ変換アルゴリズムは有効であることが示された。  
キーワード コミュニケーションロボット、ジェスチャ変換

## Realization of Gesture Contents Sharing among Heterogeneous Robots

Kenshiro HIROSE<sup>†</sup>, Hideyuki KAWASHIMA<sup>††</sup>, Satoru SATAKE<sup>†††</sup>, and Michita IMAI<sup>††††</sup>

<sup>†</sup> IBM Japan

<sup>††</sup> University of Tsukuba, Graduate School of Systems and Information Engineering

<sup>†††</sup> Keio University, Graduate School of Open and Environmental Systems

<sup>††††</sup> Keio University, Department of Information and Computer Science, Faculty of Science and Engineering

E-mail: <sup>†</sup>{hirose,satake}@ayu.ics.keio.ac.jp, <sup>††</sup>kawasima@cs.tsukuba.ac.jp, <sup>†††</sup>michita@ics.keio.ac.jp

**Abstract** The purpose of this paper is to share gesture contents among heterogeneous robots. In this paper, we classified gesture in 2 types; pointing gesture and track gesture. Criteria of classification is factors which are essential for each gesture. Pointing gestures are used for pointing somewhere around a robot. Which direction an arm of a robot points is essential factor for pointing gesture. Track of gesture itself is very important for track gesture. Track gestures can keep its essential factors by moving track horizontally or vertically. We made gesture translation algorithm for each classification and achieved sharing semantic information of gesture contents. We carried out two types of experiments to evaluate gesture translation algorithm. The one is quantitative evaluation by using computer simulation, and the other is qualitative evaluation by questionnaire. The results of both evaluation experiments showed that the algorithms are effective for a robot that has lost a part of joints.

**Key words** Communication Robot, Gesture Translation

### 1. はじめに

近年多く開発されている様々なコミュニケーションロボット [2], [3], [7], [9] は、現実世界に作用することによる高い表現能力と、各種センサを用いることにより、キーボードやマウス

を用いるよりも柔軟な情報入力方法を有する新しいメディアとして期待されている。コミュニケーションロボットの使い道は様々なものが考えられるが、本研究ではロボットが情報プレゼンテーションメディアとして使用される場合を考える。

この場合の問題点は、ロボットの関節構造の違いからロボッ

トがジェスチャ情報を共有できないことである．ロボットはそれぞれ関節構造が異なるために，ロボット間で単純に関節角度の情報を共有するだけでは，ジェスチャを共有できない．ジェスチャを共有できないと，ジェスチャの再利用ができないという非効率な状況が発生する．つまり，一つのジェスチャデータを全てのロボットに適用できないことが問題である．

ジェスチャ共有に関する研究には様々なものが成されている．MotionML [4] は，ロボットのモーションを共有するために，モーションキャプチャの関節情報のデータを XML の拡張言語である MotionML で記述する．関節角度を自由度の低いロボットにも共有することでジェスチャ共有を試みている．正射影に対応したジェスチャ変換 [12] は，ロボットのジェスチャを変換するために，関節の角度数だけではなく，ロボットを正面から見た際の上腕と前腕の比を考慮に入れている．前から見た前腕と上腕の比を考慮に入れることで，腕の長さの違うロボット間での見え方の差を縮めることでジェスチャ情報を共有しようとしている．

しかし，どれもジェスチャの意味情報を反映するという視点がなく，共有の際に重視すべき情報を拾いきれていない．個々のジェスチャの意味情報とは，それぞれのジェスチャ毎に異なる各ジェスチャの本質を表す特徴である．例えば，会釈のジェスチャの場合，首や腰が鉛直方向に行き来するのが重要であるが，指差しの場合は，最後に手先がどちらを向いているかが重要である，ということである．MotionML はモーションキャプチャのデータの角度情報を，多種類のヒューマノイドにおいて忠実に再現することを目的としている．正射影に対応したジェスチャ変換 [12] は，全てのジェスチャを特定の角度から見た際の見え方に注目している．つまり，どちらの研究も，ロボットの腕の見え方であったり，関節情報であったり一つの種類の特徴のみを扱っており，個々のジェスチャにそれぞれ存在する本質的な特徴を扱っていない．

本研究では，ロボットの意味情報を共有するために，ロボットのジェスチャを直示ジェスチャと軌道重視ジェスチャに分類した．さらに，それぞれの分類毎にジェスチャの変換アルゴリズムを作成した．直示ジェスチャは，ある目標物をロボットが指差すジェスチャである．直示ジェスチャの特徴は，ジェスチャの軌道ではなく，ジェスチャ到達点の正確性にある．そのために，本研究ではインバースキネマティクス [6] を駆使したアルゴリズムを作成した．軌道重視ジェスチャは，到達点ではなく，ジェスチャの軌道自体が重要なジェスチャである．軌道重視ジェスチャは，描写ジェスチャとその他のジェスチャの 2 種類に分かれ，それぞれ保持すべき軌道の特徴が異なる．そのため描写ジェスチャとその他のジェスチャで別のアルゴリズムを提案する．

本論文の構成は以下の通りである．2. 節にて本論文で提案するジェスチャ共有機構を述べる．3. 節にて提案機構の評価を述べる．最後に 4. 節にて結論を述べる．

## 2. ジェスチャ変換手法の提案

複数のロボットでジェスチャを共有するには，ロボット X に

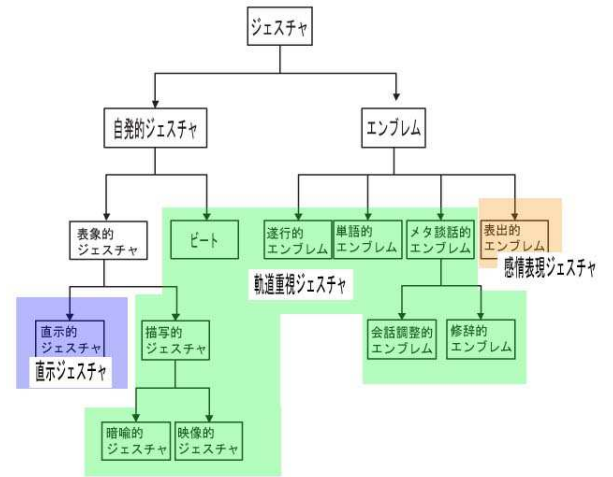


図 1 ジェスチャ分類

Fig. 1 Gesture Classification

おけるジェスチャをロボット Y で動作するように変換する必要がある．本節では，本研究におけるジェスチャの分類を述べる．本研究では，文献 [11] で分類されている人間のジェスチャを，ロボットのジェスチャを変換する際に重視すべき特徴は何かという観点から 3 種類に分類し直した．図 1 にその分類を示す．

各分類について説明する．まず，どこかを指すという動作は軌道ではなく最後のポーズが重要なので，直示ジェスチャを一つのカテゴリとした．そして，最後のポーズではなく，手先や肘の軌道自体が大きな意味を持つジェスチャに関しては軌道重視ジェスチャに分類した．そして感情表現は，軌道だけではなく，手足のモーターの速度や速度の変化が重要な要素なので，別の分類とした．ただし，本論文では感情表現ジェスチャ以外の 2 種類のみを扱う．

変換アルゴリズムには二つの段階がある．

(1) 入力 — 変換元のジェスチャデータから，アルゴリズムを適用するために必要な特徴を得る段階．入力の段階では，ユーザが入力する情報と，ジェスチャから導き出される情報の 2 種類の情報を得る．

(2) ジェスチャ変換 — 入力の段階で抽出された特徴を駆使して，変換先のロボットの関節情報を生成する段階以下，アルゴリズムについて説明する際には，この二つの段階に注目して述べていくことにする．

### 2.1 直示ジェスチャ

直示ジェスチャとは，ロボットが空間上のとある点を指すジェスチャである．前述したように，最後のポーズに至るまでの手足の動きは関係ない．最後のポーズがいかにそちらの方向を指しているかがポイントのジェスチャである．

#### 2.1.1 入力

直示ジェスチャのアルゴリズムでは，変換元のジェスチャから 3 種類の情報を入力として使用する．

- (1) 指差しに使っている腕
- (2) 差している方向
- (3) 腕の縮み方

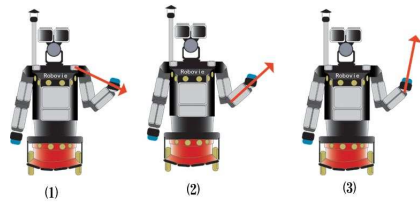


図 2 方向ベクトル  
Fig. 2 Direction Vector

指差しに使っている腕は、ジェスチャ作成者が自分で選択する。右腕と左腕のどちらかを選択す直示ジェスチャの際にユーザが入力する情報はこれだけである。指している方向と指している距離の 2 点については、ジェスチャの内容から算出する。

ジェスチャの差している方向の算出方法について説明する。ジェスチャの差している方向については、三種類のベクトルを算出している。

(1) 肩から手首まで — 指差しに使っている腕の肩と手首の関節を結んだベクトルを算出し、これを以下「肩-手首ベクトル」と呼ぶことにする (図 2-(1))。

(2) 肘から手首まで — 指差しに使っている腕の肘から手首の関節を結んだベクトルを算出し、これを以下「肘-手首ベクトル」と呼ぶことにする (図 2-(2))。

(3) 手首から手先まで — ジェスチャを作成したロボットに手首から先の関節がある場合、指差しに使っている腕の末端とその手前の関節を結んだベクトルを算出し、これを以下「手首-指先ベクトル」と呼ぶことにする (図 2-(3))。

腕の縮み方とは、ロボットの腕を完全にまっすぐに伸ばした状態と比べて、手の先がどれだけ身体の近くにあるかということである。腕の長さを  $A$  として、ロボットの肩から腕の先までの距離を  $D$  とすると、腕の縮み率  $S$  は式 1 で表される。

$$S = D/A \quad (1)$$

腕の縮み方を入力として扱った理由は、指差しにおいて腕の縮み方という情報は、以下のことを示すからである。

- 対象物までの距離 — 腕をより畳んでいると、近くのものを目指している
- 物理的制約 — 腕を完全に伸ばしてしまうと、ロボットの腕が周りに当たってしまう
- 控え目に指差しをさせたい — 大きな動作で指差しを行うのは、屋内での案内業務等、落ち着いた雰囲気が求められるタスクにおいて、落ち着かない印象を与える。

以上のようなジェスチャ作成者の意図を反映させるために、腕の縮み率を考慮にいれることにした。これらの意図というのは明示的にユーザから示されるわけではなく、筆者が直示ジェスチャの変換アルゴリズムを考える上で想定しているユーザ側の意図である。

### 2.1.2 ジェスチャ生成

3 種類の入力された情報から、どのように変換先のロボットで角度情報データを生成するのかについて説明する。角度情報データの生成は、3 ステップで行われる。

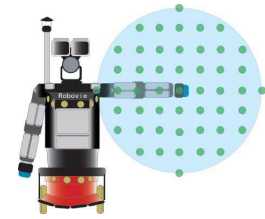


図 3 指差し候補点  
Fig. 3 Pointing Candidate

### (1) 指差しに使う腕の決定

「指差しに使っている腕」の情報から、指差しに使用する腕を決定する。そして指差しに使っていない方の腕は、まぎらわしさを回避するために関節の動きを 20%<sup>(注1)</sup>に制限する。

### (2) 指差し候補点の算出

次に、指差しに使用する腕の肩から半径  $A$  の球内に、 $A \times 0.1$  間隔で指差し候補点を算出する。指差し候補点が算出された様子を図 3 に示す。

指差し候補点が決定したら、それぞれの候補点について以下の二つの条件でインバースキネマティクスを適用する。そしてインバースキネマティクスが成功した点を全て指差し候補点とする。

- RPY(roll,yaw,pich)を「肘-手首ベクトル」として適用する。± 10 度までは誤差があっても良い
- 肩から「肩-手首ベクトル」方向の直線からの距離が  $A \times 0.1$  未満の点で、RPY はどうなっても良い

### (3) 指差し点の算出

指差し候補点の中から最終的に指差し点を算出する方法について述べる。最終的なポイントは、肘と手先の 3 次元の空間座標が変換元のジェスチャと比べて一番近いものを選択する。ジェスチャの変換元のロボットと、変換先のロボットで関節構造が異なる場合は、消失している関節を除いて考える。

## 2.2 軌道重視ジェスチャ

軌道重視ジェスチャは、さらに細かく物体描写ジェスチャと軌道範囲ジェスチャに分類される。物体描写ジェスチャは、描写ジェスチャの一部であり、物体の輪郭や図形の描画を意味する。軌道範囲ジェスチャは、軌道重視ジェスチャの残りの部分を占めている。

### 2.2.1 物体描写ジェスチャ

物体描写ジェスチャとは、空間上に何らかの物体や抽象的な表現を描くジェスチャである。ジェスチャ変換の視点から見ると、描写ジェスチャというのは、手先が描く軌道の形を保つことが重要である。例えば、図 4-(1)にあるように、正方形を描きたかったような場合でも、軌道が斜めになってしまうと、菱形になってしまいその意味が失われてしまう。また、軌道の一ヶ所が欠けてしまうと、その意味が著しく失われてしまう (図 4-(2))。

従って、物体描写ジェスチャの変換アルゴリズムには (1) 軌道の形を回転させない、および、(2) 軌道の一部を欠落させな

(注1): 今後出現するこれらの数字は経験的に決められている

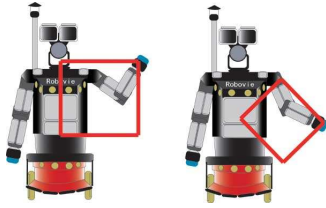


図 4 斜めになった軌道  
Fig. 4 Leaning Orbital

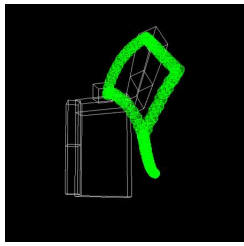


図 5 3次元座標のプロット

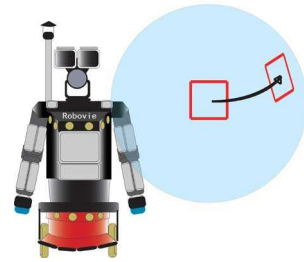


図 6 軌道の平行移動

Fig. 5 Plot of 3D point Fig. 6 Orbit Translation

い、の 2 特徴を保持することが重要である。

a) 入力

物体描写ジェスチャの変換アルゴリズムにおける入力は、ジェスチャ実行中における一定時間毎の末端関節の 3 次元座標である。3 次元座標は、シミュレーション上で実際にジェスチャを実行させ、一定フレーム数毎に座標をプロットしていくことで求める。シミュレーション上で末端関節の 3 次元座標がプロットされる様子を、図 5 に示す。

b) ジェスチャ生成

物体描写ジェスチャのアルゴリズムは、ジェスチャ変換先のロボットの表現能力が低下しても 2.2.1 節で述べた 2 点の特徴を保持する必要がある。そのため、物体描写ジェスチャにおけるジェスチャ変換のアルゴリズムは、軌道全体に、(1) 水平移動、(2) 垂直移動、(3) 前後移動、(4) 縮小、を施すことになる。3 種類の平行移動の操作は、肩を中心とした球面上を平行移動する (図 6)。

ロボットがジェスチャを行う際、それを見る人間は、正面からそれを見ている可能性が高い [12]。よって、ジェスチャの平行移動は、なるべく人間の視界から外れないように行う必要がある。描写ジェスチャの変換アルゴリズムに関する具体的な手順について説明する。

まず最初に、算出されたプロット点それぞれにインバースキネマティクスを適用し、変換先のロボットが手を伸ばせない点を求める。その後、水平方向、垂直方向、前後方向にそれぞれ少しずつ移動させ、軌道の縦横幅を縮小させる。その具体的な手順を図 7 に示す。正常終了すると、計算された軌道が解になり、異常終了した場合は、最も到達不能点が少なかった点が解となる。

2.2.2 軌道範囲のジェスチャ

軌道範囲ジェスチャとは、遂行的エンブレム、メタ談話的エンブレム、単語的エンブレム、の 3 種類のジェスチャに描写

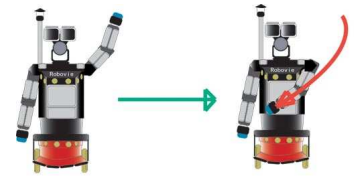
```

1: while (1){
2:   if (水平方向の移動角度 < 20 )
3:     到達不能点が多い側と逆方の水平方向に 5 度移動;
4:   if (到達不能点が全てなくなる)
5:     正常終了;
6:   if (垂直方向の移動角度 < 20 )
7:     到達不能点が多い側と逆方の水平方向に 5 度移動;
8:   if (到達不能点が全てなくなる)
9:     正常終了;
10:  前方向に、前方向に、A × 0.1 軌道を水平移動;
11:  if (到達不能点が全てなくなる)
12:    正常終了;
13:  if (軌道のサイズが元々の 70%以上)
14:    軌道のサイズを 5%縮小;
15:  if (到達不能点が全てなくなる)
16:    正常終了;
17:  if (垂直方向、水平方向の移動角度が共に 20 度以上 & 軌道サイズが元の 70%以下)
18:    異常終了;
19: }

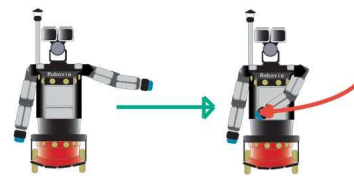
```

図 7 物体描写ジェスチャアルゴリズム

Fig. 7 Algorithm for Object Drawing Gesture



(1)



(2)

図 8 投げるジェスチャ

Fig. 8 Throwing Gesture

ジェスチャの一部を加えたものである。描写ジェスチャの一部とは、話の登場人物の動作を真似するジェスチャであり、具体例としては「投げる」や「打つ」がある (以下このジェスチャを動作描写ジェスチャと呼ぶ)。軌道範囲ジェスチャをジェスチャ変換の観点から見た際の特徴は、ジェスチャの軌道の動く範囲に特徴があるということである。

動作描写ジェスチャを例にとって説明する「投げる」というジェスチャは、左手の先が身体の左後ろから斜めに身体の前まで大きな軌跡を描いている (図 8-(1))。この「投げる」ジェスチャに関して、軌道の範囲が変わると大きく意味が失われることとなる。図 8-(2) にあるような腕が上まであがり切らない状態では、投げているというよりも、手を水平移動させただけのように見える。



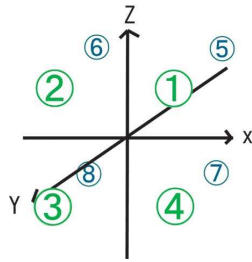


図 9 象限のナンバリング  
Fig. 9 Numbering of Quadrant

#### a) 変換アルゴリズム

本節では軌道範囲ジェスチャの変換アルゴリズムについて説明する。このアルゴリズムでは、軌道範囲ジェスチャの、ジェスチャの際に手先や肘の動く範囲がジェスチャの重要な特徴であるという点を反映する必要がある。

そのため、本研究では身体のある点（例えば肩）を中心に、周りの空間を上下左右前後の八象限に分け、ジェスチャの軌道において身体の別の点（例えば手首）が八象限のどこを通るのかを重視するアルゴリズムを提案する。以下、説明を簡単にするために八象限のナンバリングについて説明する。ロボットから見て左方向に  $x$  軸、上方向に  $y$  軸、前方向に  $z$  軸を置き、八象限を以下のように定義する。

- $x > 0, y > 0, z > 0$  の領域を第一象限とする (1)
- $x < 0, y > 0, z > 0$  の領域を第二象限とする (2)
- $x < 0, y < 0, z > 0$  の領域を第三象限とする (3)
- $x > 0, y < 0, z > 0$  の領域を第四象限とする (4)
- $x > 0, y > 0, z < 0$  の領域を第五象限とする (5)
- $x < 0, y > 0, z < 0$  の領域を第六象限とする (6)
- $x < 0, y < 0, z < 0$  の領域を第七象限とする (7)
- $x > 0, y < 0, z < 0$  の領域を第八象限とする (8)

八象限とロボットの関係を図示したものを図 9 に示す。

#### b) 入 力

軌道範囲ジェスチャの変換アルゴリズムにおける入力、は、四種類である。以下、それぞれの取得方法を述べる。

- プロットの到達不能点 — 軌道上の点全てにインパースキネマティクスを適用し、到達不能点を算出する
- 軌道範囲を定義する際の原点と末端の点 — 手首の動きと肘の動きを計算し、八象限の原点と末端を決定する
  - 肘がほとんど動いていない場合は、肘を中心として末端を手首とする。
  - 肘が大きく動いている場合は、肩を中心として末端を手首とする。
- 八象限を行き来する順番 — 末端関節が八象限をどのような順番で遷移していったかを算出する。
- 八象限を通過する際の座標 — また、象限を移動する際に、境目となる平面を通過した際の座標を算出する。

#### c) ジェスチャ生成

続いて、変換元ジェスチャで算出された値を変換先ロボットに当てはめる際の手順について説明する。軌道を近づけたり遠

```

1: for (全ての到達不能点) {
2:   if (点を消しても象限を割らない)
3:     点を消去;
4:   else {
5:     軌道を遠近移動して象限を越えない点を調査;
6:     if (象限を割らない点が見つかった)
7:       その軌道を採用;
8:     else
9:       その点を削除;
10:  }

```

図 10 軌道範囲ジェスチャの変換アルゴリズム  
Fig. 10 Algorithm to Translate Orbit Range Gestures

ざげたり、削ったりしながら到達不能点を少なくしていく。このアルゴリズムを図 10 に示す。

### 3. 評 価

本章では、本論文にて提案したジェスチャ変換機構について行った評価について述べる。評価として定量的評価と定性的評価の二種類を行った。実験では、Robovie-2 [3] と Robovie-M [1] の関節モデルをやや変更したものにもジェスチャを実行させた。即ち、実質的にジェスチャ変換機構を 4 種類のロボットについて実装したことになる。以下に 4 種類のロボットの詳細を示す。

- Robovie-M: 全関節使用 (A1)
- Robovie-M: 関節可動範囲を半分に制限 (A2)
- Robovie-2: 全関節使用 (B1)
- Robovie-2: 肘関節を固定 (B2)

また、評価における比較対象は、角度情報のみを共有した場合とする。これら 4 種類のパターンから、ジェスチャ変換元のロボットと、ジェスチャ変換先のロボットの 2 種類のロボットを選択しなければならない。

ジェスチャ変換アルゴリズムは、表現能力の高いロボットで作成されたジェスチャを表現能力の低いロボットに対して移植する場合を想定しているため、図 11 に示したパターンで変換を行うことにする。このような組合せにした理由は、Robovie-M は最も関節の自由度が多く、表現能力が高いからである。Robovie-2 は、Robovie-M から腰関節を消失させたロボットである。よって今回の評価に使用した 4 種類のロボットは、以下のように言い換えることができる。

- A1 — 変換元のロボット
- A2 — 同じロボットで手足の関節の限界角度が半分のもの
- B1 — 違うロボットで関節は腰関節がなくなっているもの
- B2 — 違うロボットで関節は肘関節と腰関節がなくなったもの

評価対象のジェスチャは、直示ジェスチャ 3 種類、軌道重視ジェスチャ 4 種類の合計 7 種類である。その詳細を表 1 に示す。

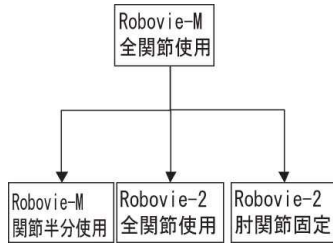


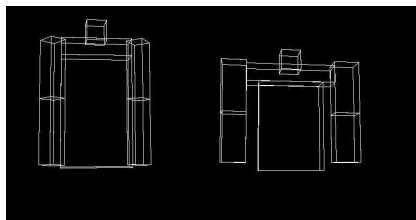
図 11 ジェスチャ変換の組合せ

Fig. 11 Combination of Gesture Translations

表 1 実験に使用したジェスチャ

Table 1 Gestures used in Experiments

ジェスチャの種類	詳細
直示ジェスチャ	左を指す
	左上前方を指す
	左上後方を指す
軌道重視ジェスチャ	バイバイ
	投げる
	手招き
	菱形を描く



(1) (2)

図 12 CG 上で実装されたロボット

Fig. 12 Robots Implemented on CG

### 3.1 定量的評価

定量的評価は、OpenGL を使用したシミュレーションで行った。評価の基準は、アルゴリズム適用時と、角度情報変換時アルゴリズム適用時と、関節角度のみ共有時のジェスチャが、どれだけ元々のジェスチャと離れているのかを定量的に評価した。また、インバースキネマティクスを提案手法であるジェスチャ変換機構に組み込むにあたって、ライブラリとして ROBOOP [8] を採用した。ROBOOP とは、ロボットアームのシミュレーションのためのライブラリであり、C++ で全て記述されている。

#### 3.1.1 定量的評価の実験環境

定量的評価の実験の環境について述べる。実験は Linux マシン上で OpenGL を利用し、CG 上でワイヤーフレームのロボットを実装した。ロボットの外見を図 12 に示す。それぞれ、図 12-(1) が Robovie-2 で、図 12-(2) が Robovie-M を CG 上で再現したものである。

シミュレーションの 1 フレームは 1/20 秒である。この CG ロボットは、以下の値を変数としてロボットの各パーツを再現できるようになっている。現実のロボットは、Robovie-M と Robovie-2 でかなりサイズに差があるので、両方のロボットの腕の長さを 5 として大きさを揃えた。

表 2 定量的評価の結果

Table 2 Results of Quantitative Evaluation

ジェスチャ名	変換先ロボット	角度情報共有	アルゴリズム適用
左を指す	A2	1.02	0.94
左上後方を指す	A2	4.07	2.28
左上前方を指す	A2	3.91	1.82
バイバイ	A2	2.23	1.83
菱形を描く	A2	3.10	2.48
手招き	B1	4.12	2.26
投げる	B1	3.74	1.10
左を指す	B1	3.29	0.90
左上後方を指す	B1	4.22	1.29
左上前方を指す	B1	4.82	1.40
バイバイ	B2	3.59	1.97
菱形を描く	B2	2.76	1.84
手招き	B2	3.81	1.40
投げる	B2	3.26	1.21

表 3 ジェスチャ種類別の結果

Table 3 Results by Gesture Types

	角度情報のみ共有	アルゴリズム適用
直示ジェスチャ	3.56	1.43
軌道重視ジェスチャ	3.32	1.76

- 存在する関節 — 定義されている関節のうち、ロボットに存在する関節
- 手足、胴体、頭の大きさ — 縦、横、奥行きの 3 つを指定して調整する。各パーツは直方体で描かれる
- 関節の限界角度数 — 各関節が何度まで曲るのか

#### 3.1.2 定量的評価の実験方法・結果

定量的評価の実験方法について述べる。3.1.1 節において述べた環境で、CG ロボットにジェスチャを実行させる。ロボットの手の座標のユークリッド距離 [10] をとり、どれだけアルゴリズム適用時の方が変換元と比べて近い軌跡を辿っているのか算出する。算出の方法は、直示ジェスチャか、軌道重視ジェスチャかで異なる。感情表現ジェスチャに関しては、関節の軌道は軌道重視ジェスチャと同じため、軌道重視ジェスチャの中に入れて考える。

- 直示ジェスチャ — 最後の到達点の手首座標のユークリッド距離を算出する
- 軌道重視ジェスチャ — 0.2 秒毎にロボットのスナップショットをとり、各スナップショットにおける手首座標のユークリッド距離を算出し、その平均を算出する

以上の方法で算出された値を、表 2 に示す。表 2 の結果をジェスチャの種類毎にまとめたものを表 3 に、適用先ロボット毎にまとめたものを表 4 にそれぞれ示す。

#### 3.1.3 定量的評価の考察

本節では、定量的評価の結果に関する考察について述べる。CG のロボットは両方共、腕の長さが 5 となるように最適化されているため、直示ジェスチャ、軌道重視ジェスチャ共に平均して腕半分弱の距離の差があったことになる。

表 4 変換先ロボット種類別の結果

Table 4 Results by Translation Targets

	角度情報のみ共有	アルゴリズム適用
A2	2.86	1.87
B1	4.04	1.39
B2	3.36	1.61

最も大きく差が出たのは、Robovie-M が腰の関節を使用したジェスチャを行った際である。腰の関節は、他のどの関節よりも最も腕に近いために、手先座標への影響が最も大きかったものと思われる。一方で、ほとんど差が出ないタイプのジェスチャも存在した。限界角度を半分にしたロボットへ「左を指す」のジェスチャを移植した際は、到達点の差がほとんどゼロに近かった。これは、もともとのジェスチャが、おおよそ左を指していたことから、アルゴリズムを適用しても差がでなかったものである。そして軌道重視ジェスチャの一部では、むしろ角度情報を共有するだけよりも軌道差が大きくなったものがあった。肘を固定したロボットに対して「投げる」のジェスチャを適用した場合がそれに当たる。

しかし、結果にあるように平均すると腕半分の差が直示ジェスチャ、軌道重視ジェスチャの両方に出たので、定量的評価の点では本研究の手法は効果を上げたと言える。

### 3.2 定性的評価

定性的評価としては、アルゴリズム適用時と、関節角度のみ共有時のジェスチャを実際にロボットに実行させ、どちらの方が変換元に似ているかを人間に主観的に判断してもらう。

#### 3.2.1 定性的評価の方法・結果

定性的評価の実験方法について述べる。この実験では、各ジェスチャを、それが何のジェスチャなのかキャプション付きで見てもらい、その結果、変換前ジェスチャと似ているのは、「アルゴリズム適用時」か「角度のみ共有時」か、それとも「どちらとも言えない」かの3択で選んでもらう。また、それぞれのジェスチャについて何かコメントがあれば、それも添えてもらうようにした。キャプション付きで動画を見せる理由は、ジェスチャを判断する人間が、そもそもそれが何のジェスチャなのか理解していないと、そのジェスチャの意味情報が正しく保存されているかどうか判断できないからである。

定性的評価の実験結果をジェスチャの種類別にまとめたものを表6に、適用したロボット別にまとめたものを表7に示した。

#### 3.2.2 定性的評価の考察

本節では、定性的評価の結果に関する考察について述べる。定性的評価の結果、表6にあるようにジェスチャの種類別に見ると、6割以上がアルゴリズム適用時の方が似ていると回答したため、どちらのカテゴリに関しても、本研究の手法は成果を上げたと言える。

一方で、変換先ロボット別の結果(表7)を見ると、関節の限界角度を半分にして Robovie-M に適用した結果だけ、他の2項目とほとんど差がないのが分かる。つまり、人間から見た場合、角度半分のロボットに適用した場合は全く差がつかない場合が多く、むしろ角度情報のみを共有した場合の方が似ている

表 5 定性的評価の結果

Table 5 Results of Qualitative Evaluation

ジェスチャ名	変換先ロボット	角度情報共有	アルゴリズム適用	判別不能
左を指す	A2	4	2	6
左上後方を指す	A2	0	9	3
左上前方を指す	A2	7	2	3
バイバイ	A2	3	6	3
菱形を描く	A2	3	3	6
手招き	B1	4	7	1
投げる	B1	2	10	0
左を指す	B1	0	10	2
左上後方を指す	B1	0	11	1
左上前方を指す	B1	0	10	2
バイバイ	B2	0	12	0
菱形を描く	B2	4	7	1
手招き	B2	0	7	5
投げる	B2	1	9	2

表 6 ジェスチャの種類別の結果

Table 6 Results by Gesture Types

	角度情報のみ共有	アルゴリズム適用時	判別不能
直示ジェスチャ	15.3%	61.1%	23.6%
軌道重視ジェスチャ	16.8%	64.2%	18.8%

表 7 変換先ロボット別の結果

Table 7 Results by Translation Target

	角度情報のみ共有	アルゴリズム適用時	判別不能
A2	27.1%	37.3%	35.6%
B1	10.0%	80.0%	10.0%
B2	10.4%	72.9%	16.7%



(1) (2)

図 13 左上前方を指すジェスチャ

Fig.13 Left Upper Front Pointing Gesture

と回答する被験者も多数いた。

ここで、限界角度半分のロボットに対して「前方左上を指す」ジェスチャ変換を適用した際のアンケートに添えてあったコメントを以下にいくつか挙げる。

- どちらも手の上げ方が明らかに違う。
- ほぼ同じに見える。
- 肘から先が曲っているのがやや不自然に見えた

実際に「左を指す」を被験者に見せた際の映像を図13に示す。図13-(1)が角度情報のみ共有時であり、図13-(2)がアルゴリズム適用時である。

アンケートの内容から読み取れるように、被験者の中にはむしろ、アルゴリズムによる微調整が不自然に感じるという者がいる。つまり、各関節の限界角度が低い場合にはジェスチャ変換アルゴリズムを適用しても、人間から見た場合十分な効果があがらないということが分かる。

また、前述した通り、今回は「どちらが変換元のジェスチャと似ているか」という聞き方をしたので、「何をもって似ているのか」という基準が被験者ごとに異なることがあった現象も興味深かった。例えば、「菱形を描く」のジェスチャを例にとってみると、「1番のジェスチャの方が変換元と似ているが、2番のジェスチャの方がより菱形を描いているように見えた」といった回答が寄せられた。

### 3.3 考察

本節では、定性的、定量的の両方の評価を踏まえた上での考察をする。定量的評価の結果から、手先の物理的位置という観点からは、ジェスチャ変換アルゴリズムが十分に機能することが示された。定性的評価の結果では、関節の自由度が減少する場合には十分に対処できた。しかし、限界角度半分のロボットに対しては全く効果が得られなかった。

定量的評価、定性的評価の両方を踏まえ、以下の3点が得られる。

- 関節の自由度が減った場合には、その他の関節をさらに回すことで、元々のジェスチャと類似した印象を与えるジェスチャを行うことができる。
- 関節の限界角度が全体的に低くなってしまうと、ジェスチャの意味情報を保つことができない。3次元座標の位置が近くても、人間の印象としては不自然に感じてしまうことが多い。
- 人間の目から見て違いが分かるためには、定量的にかなり差があることが前提であると考えられる。なぜならば、「左を指す」のように、定量的評価で結果の出なかったジェスチャに関しては、例外なく定性的評価でも結果が出なかったという現象が見られたからである。

## 4. 結論

本研究は、異種ロボット間でジェスチャ情報を共有するためのシステムであるジェスチャ共有機構を提案した。ジェスチャ共有機構の課題は、ジェスチャの意味情報を共有することであった。

ジェスチャ共有機構では、上記問題点を解決するために、ジェスチャ変換機構においてジェスチャを2種類に分類した。その2種類とは直示ジェスチャと軌道重視ジェスチャだった。直示ジェスチャはどこかを指差すジェスチャであり、最後のポーズが本質的な特徴である。軌道重視ジェスチャは、ジェスチャの軌道自体に意味があり、軌道を並行移動させたり縮小した場合でも本質的な特徴を保存できる。それぞれの分類毎にジェスチャの変換アルゴリズムを作成したことで、ジェスチャの意味情報の共有が実現できた。

ジェスチャ変換アルゴリズムを評価するために、2つの評価実験を行った。1つ目は評価シミュレーションを用いた定量的評価であり、2つ目はアンケートによる定性的評価である。両

実験の結果、関節を消失したロボットに対しては、ジェスチャ変換アルゴリズムは極めて有効であることが示された。

図1で分類したが本論文では扱わなかった感情表現ジェスチャの生成が今後の課題である。感情表現ジェスチャをジェスチャ変換の視点から見ると、ジェスチャの軌道だけではなく、その軌道の上をどのような速度やリズムで動くかが重要であることが分かっている[5]ものの、我々の知る限りその変換技法は未だ実現されていない。

## 文献

- [1] *Robovie-M*.  
<http://www.vstone.co.jp/top/products/robot/Robovie-M.html>.
- [2] *AIBO*. <http://www.jp.aibo.com>.
- [3] 神田崇行, 石黒浩, 小野哲雄, 今井倫太. 人間と相互作用する自律型ロボット Robovie の評価. 日本ロボット学会誌, Vol. 44, No. 11, pp. 2699–2709, 2002.
- [4] Kitagishi. Development of motion data description language for robots based on extensible markup language -realization of better understanding and communication via networks-. In *Journal of Robotics and Mechatronics*, Vol.14, No.5, pp. 471–478, 2002.
- [5] Toru Nakata. *Generation of whole-body expressive movement based on somatical theories*, 2002.
- [6] O.Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. In *IEEE Trans. Robot Automat.* Vol. RA-3, pp. 49–62, 1987.
- [7] *QRIO*.  
<http://www.sony.net/SonyInfo/QRIO/top.html>.
- [8] *ROBOOP - A robotics object oriented package in C++*.  
<http://www.cours.polymtl.ca/rooop/>.
- [9] *wakamaru*. <http://www.sdia.or.jp/mhikobe-e/products/etc/robot>
- [10] 川島英之, 今井倫太, 遠山元道, 安西祐一郎. センサデータベースシステム KRAFT の設計と実装. 情報処理学会論文誌: データベース, Vol. 45, No. SIG14(TOD24), pp. 39–53, 2004.
- [11] 喜多壮太郎. ジェスチャー考えるからだ, 2002.
- [12] 平松雅巳, 八木康史, 橋本浩一, 谷内田正彦. 対話者方向からのアピラランスに基づくロボットアバターのためのジェスチャ生成, 2001.