

RSV：コミュニケーションロボットのための センサデータ解析支援システム

中村 学[†] 川島英之^{††} 佐竹 聡[†]
今井倫太^{††} 安西 祐一郎^{††}

コミュニケーションロボットの行動プログラムを設計するにはインタラクションを行う人間の行動を予測する必要があるが、これは困難であるため効率的に実現されてこなかった。この問題を解決するために本論文はセンサデータ解析支援システム RSV を提案する。RSV は従来研究では困難であった膨大な標本の比較による人間行動発見を可能にするだけでなく、人間行動とセンサデータを対応付けることで、発見した人間行動を容易に行動プログラムへ組み込むことを可能にする。本研究の有用性を評価するために、コミュニケーション実験における人間行動を RSV を用いて分析した。その結果、従来の行動プログラムでは想定外の人間行動を検出できた。

RSV: Sensor Data Analysis System for Communication Robot

MANABU NAKAMURA,[†] HIDEYUKI KAWASHIMA,^{††} SATORU SATAKE,[†]
MICHITA IMAI^{††} and YUICHIRO ANZAI^{††}

The expectation of user behaviors is essential for programming the actions of communication robots. However, efficient supports for the expectations have not been realized yet. To tackle this problem, this paper presents a novel sensor data analysis system RSV. RSV enables to discover user behaviors by comparing enormous data samples. RSV associates user behaviors with the sensor data and it enables to incorporate detected behaviors into robots' action programs. To evaluate RSV, we conducted communication experiments, obtained the data samples of user behaviors and analyzed them by using RSV. The result of experiments showed that RSV could detect the features of user behaviors which were not considered in the current action programs on communication robot Robovie.

1. 序 論

コミュニケーションロボットは人間とロボットが共生するという目的で作られたロボットであり、その行動は行動プログラムで生成される。この行動プログラムでは、センサデータを利用して人間行動を認識する。人間行動によるセンサデータの変化は設計者が考えるだけでは想定しきれず、実際にプログラムを動かして発生するセンサデータを調べなければならない。それゆえ、実際に人間とロボットをコミュニケーションさせて得たデータから人間行動を分析する必要がある^{1),2)}。そして、設計者が想定しきれなかった人間行

動を行動プログラムに反映させることが求められる。

人間とロボットのコミュニケーションから得たデータの解析を補助する目的で、R-support³⁾ というソフトウェアが提案された。R-support はセンサデータの可視化を行い、ビデオデータに対応するセンサデータを閲覧可能にすることで、センサデータと人間行動の対応付けを実現した。

R-support はビデオデータとセンサデータの対応を対象としたため、個人におけるセンサデータと人間行動の対応を知るには有効である。しかし、行動プログラムは一般的な人間行動に対応して作られるものである。R-support が可能な分析は、“人間 A は頭を触れる”、“人間 B は肩を触れる”、“人間 C は腕を触れる”などの個別分析であり、“一般に人間は頭を触れる傾向がある”などの一般的分析を行えない。それゆえ、行動プログラムを修正するには、多数標本の比較によるセンサデータの傾向分析が不可欠となり、莫大な労力を求められる。

[†] 慶應義塾大学大学院理工学研究科開放環境科学専攻
School of Science for OPEN and Environmental Systems, Faculty of Science and Technology, Keio University

^{††} 慶應義塾大学理工学部情報工学科
Department of Information and Computer Science, Faculty of Science and Technology, Keio University

そこで本研究では、センサデータの傾向を分析するフレームワークと、そのフレームワークに基づくデータ解析支援ソフトウェア RSV を提案する。

本研究で提案するフレームワークは以下の5つの手順から構成される。1. センサデータから解析を行う標本を多数集める。2. 集めた標本のセンサデータを可視化する。3. 可視化データから傾向を発見する。4. 傾向が見られたデータ部分と同時刻のビデオデータを閲覧する。5. ビデオデータから実際の間行動を確認する。この手順では多数標本を比較してセンサデータの傾向を把握した後にビデオデータを閲覧することで、一般的な人間行動とセンサデータの傾向を対応可能にする。

RSV は取得したセンサデータから、解析に要する多数の標本を集めるデータ選択機能を提供する。RSV は集めた標本のセンサデータを可視化する機能を提供し、解析者によるデータ傾向の発見を補助する。そして、RSV は傾向が発見されたデータ部分について、ビデオデータを再生させる機能を提供する。RSV はデータ選択機能でラベル付けされたセンサデータを選択させることで、膨大なセンサデータから必要部分を抜き出せる。RSV は複数データの一括したグラフ化と、グラフのスクロールを用いて多数の標本を扱える可視化を提供する。RSV はこれらの機能を用いて、一般的な人間行動とセンサデータの傾向を対応可能にし、行動プログラム設計を支援する。

提案フレームワークと RSV の有用性を示すために、6人の被験者にロボットと実際にコミュニケーションをさせてセンサデータを取得し、データ解析・人間行動分析を行った結果を示す。

本論文の構成は以下のとおりである。2章でコミュニケーションロボット Robovie とソフトウェアの要件について述べる。3章で関連研究を述べる。4章で人間行動分析フレームワークと RSV を提案する。5章で RSV の設計を述べる。6章で RSV の実行例を示し、7章で RSV の議論を行い、8章で結論を述べる。

2. 背景

2.1 コミュニケーションロボット Robovie

コミュニケーションロボットは、人間とロボットが共生するというコンセプトで作られたロボットであり、音声やジェスチャで人間と実世界のコミュニケーションを行う。代表的なコミュニケーションロボットには、QRIO と ASIMO があげられる。本研究ではコミュニケーションロボット Robovie¹⁾ を対象として RSV を設計する。Robovie の概観を図1に示す。

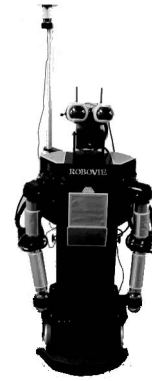


図1 コミュニケーションロボット Robovie
Fig.1 Communication robot Robovie.

2.2 センサデータ

本研究で対象とするコミュニケーションロボット Robovie は周辺環境を認識するために24個の距離センサ、16個のタッチセンサ、10個のパンパセンサを持つ。カメラデータは右目のカメラ、左目のカメラ、全方位カメラから取得できる。

Robovie は人間行動を認識するために、距離センサデータ、タッチセンサデータ、カメラデータを使用する。距離センサデータは整数値、タッチセンサデータは2値、カメラデータはjpg形式である。パンパセンサは人間行動の認識ではなく、衝突検出のために使用される。

Robovie が用いるセンサからは、大量のデータが発生する。Robovie は50個のセンサについてデータを30msec周期で取得する。Robovie を1時間稼働させたとしても、そのセンサデータの数には600万個にもおよぶ。通常の実験は30人程度に約10分行うため、約5時間3,000万個のセンサデータを取得する。したがって、センサデータから人間行動を解析するには、膨大なセンサデータから必要部分を抽出する手段が必要である。

2.3 行動プログラムの従来の開発手段

これまで、コミュニケーションロボットの行動プログラムは次の手順で開発されてきた。

- 開発手順1 設計者がロボット行動に対する人間行動を予測する。
- 開発手順2 人間行動の予測に基づいてプログラムを作成する。
- 開発手順3 設計者がロボットとインタラクションを行う。
- 開発手順4 動作確認を行い、想定した動作をしない場合は原因を分析する。想定した動作をする場合

は開発手順 6 へ進む。

開発手順 5 分析結果に基づいてプログラムを修正し、
開発手順 3 へ戻る。

開発手順 6 一般の人間とロボットをインタラクションさせる。

開発手順 7 人間行動を分析し、設計者の予測した行動が適切かを判断する。適切な場合は開発手順 9 へ進む。

開発手順 8 実際の人間行動に対応するようにプログラムを修正し、再度人間行動の分析が必要な場合は開発手順 6 へ戻る。

開発手順 9 人間行動に対応したプログラムが完成する。

ロボットとのコミュニケーションにおいて、人間は様々な行動をとるため、設計者の人間行動予測だけでプログラムを作成した場合、人間行動の見落としが発生する。見落とされた人間行動が発生した場合、それに対応するロボットの動作は定義されていないため、人間とロボットのコミュニケーションが円滑に行えなくなる。

それゆえ、コミュニケーションロボットの行動プログラムの作成は、設計者の人間行動予測に基づいてプログラムを作成するフェーズ A (開発手順 1 ~ 5) と実際の人間行動に基づいてプログラムを作成するフェーズ B (開発手順 6 ~ 9) の 2 フェーズに分けて行われる。

フェーズ A では設計者の予測行動を確認できればよいため、分析に要するデータ量は少ない。しかし、フェーズ B は一般的な人間行動を分析するため、多数の人間からデータを取得する必要がある。それゆえ、分析に要するデータ量は莫大になる。日常生活でロボットを運用するには、フェーズ B の綿密な実行が求められる。本研究はフェーズ B の効率化を対象とする。

3. 関連研究

3.1 人間行動分析手法

ロボットと人間のインタラクションを記録し、行動解析を行う最も一般的な手法はカメラの使用である。ビデオによる解析は、客観的な視点での人間行動解析が可能である。人間の動作をモーションキャプチャで記録し、人間行動を解析する方法もある⁴⁾。

これらの解析は人間行動の分析が目的であり、行動プログラムの修正を目的としていない。それゆえ、ロボット行動に対する人間行動を設計者が予測する際には役立つが、センサデータを直接知ることはできな

いため、人間行動とセンサデータの傾向を対応付けられない。また、多数の人間についてのビデオデータやモーションキャプチャデータを閲覧するには莫大な時間を要する。

R-support はロボットのセンサデータとビデオデータを利用して、人間行動を解析する。R-support はビデオデータとセンサデータを対応させながら、閲覧・解析する機能を有するため、人間行動とセンサデータの傾向を対応付けて解析できる。そのため、R-support は人間行動解析に有用である。

しかし、R-support は多数の標本についてビデオデータとセンサデータを対応させて解析できないため、多数の人間についての比較解析を行えず、一般的な人間行動を抽出することが難しい。また、膨大なセンサデータから必要部分を抜き出せず、ビデオデータに対応するセンサデータの閲覧のみを行うため、ビデオデータ解析時と同様に多数の人間についての解析に莫大な時間を要する。

これら従来の分析手法は個人のデータから個人の行動分析を行っている。そのため、これらを用いて多数の人間について行動分析をする手順は次のとおりになる。

分析手順 1 個人データの解析を行い、個人の行動を分析する。これを人数分行う。

分析手順 2 個人の行動の分析結果を統合して、一般的な人間行動を導く。

分析手順 1 はデータ解析を人数分行わなければならないため、莫大な労力を要する。さらに、1人1人の分析ではノイズとされるが、全体で見ると一般的な傾向となる事象が見落とされるおそれがある。それゆえ、分析手順 2 で1人1人の分析を統合する際に情報が損失する可能性がある。たとえば、人間が短時間だけロボットの頭に触れる場合、それは全体分析では意味を持つが、個人分析ではそのデータを見てもノイズとされ不要とされる可能性が高い。

3.2 データ可視化技法

文献 5) では多変量データ系列における規則性を発見するために、多様な色相の多数の画素が織りなすテクスチャの系列として、多変量データを表現する攪拌凝縮法が述べられている。文献 6) ではユーザによるカスタマイズ可能な任意の視覚化スキームを用いて、様々な DB レコードを 3D オブジェクトとして仮想空間内に実体化する手法が述べられている。文献 7) ではファジークラスタリングによるデータ視覚化手法が述べられている。文献 8), 9) ではユーザが与える抽象的な可視化目的に応じて、自動的に可視化実現方法

を決定する手法が述べられている。

文献 5) の攪拌凝縮法はデータをかき混ぜるために、センサデータ解析支援システムでの検出に必要となる、データ例外値を損ねてしまう。文献 6) の手法は 3D フレーム生成に特化したものであり、センサデータ解析支援システムで必要なデータの微細構造理解を支援することが困難である。文献 7) の手法はグラフスペクトル法であり、本研究の対象とする膨大な量のセンサデータは非グラフ構造であるために適用が困難である。文献 8), 9) の手法はユーザが対話的に可視化を行うため、ビデオデータを含めた本研究の対象データから、必要部分の抜き出しおよび可視化処理を的確にかつ低労力で実現することは困難だと考えられる。

3.3 センサネットワークプログラミング開発環境

センサデバイスに着目してプログラム開発を支援するソフトウェアに EmStar¹⁰⁾ があげられる。EmStar はユーザ空間でデバイスファイルを定義し、センサデバイスのシミュレーションとエミュレーション環境を提供する。EmStar はセンサが必要とするインタフェースをあらかじめ提供するため、多種多様なセンサを容易に扱える。

センサネットワークの低信頼性に着目してプログラム開発を支援するソフトウェアに Sympathy¹¹⁾ があげられる。Sympathy は各センサから取得したネットワーク状態データから、センサネットワーク上で生じた障害を発見し、センサネットワークのデバッグを支援する。

センサアプリケーションの開発を支援するには、膨大なセンサデータをユーザが解析しやすい形で提供することが求められるが、これらのシステムにそのような機能はない。

4. 人間行動分析フレームワークと RSV の提案

4.1 人間行動分析フレームワーク

従来人間行動分析手法では、一般的な人間行動を分析するには莫大な労力を要した。そこで本研究では、多数の人間について同時にセンサデータを閲覧し、センサデータの傾向を抽出し、センサデータの傾向から一般的な人間行動を見つけ出すフレームワークを提案する。以下に提案するフレームワークを示す。

- 分析手順 1 センサデータから解析を行う標本を集める。
 分析手順 2 集めた標本のセンサデータを可視化する。
 分析手順 3 可視化データから傾向を見つける。
 分析手順 4 データの傾向が見られた部分についてビ

デオデータを閲覧する。

分析手順 5 ビデオデータにより人間行動を確認する。

本フレームワークでは多数の標本を可視化し、可視化されたセンサデータからデータの傾向を見つけることで、一般的な人間行動がありそうな部分を見いだす。そして、データの傾向が見られた部分についてビデオデータを閲覧し、実際の人間行動を確認することで、一般的な人間行動を分析する。

本フレームワークは、多数の人間について同時に分析するため、個別分析する従来手法に比べて少ない労力で一般的な人間行動を分析できる。また、センサデータはロボットの人間行動認識に直結するため、本フレームワークによるセンサデータを利用した人間行動分析は行動プログラムの修正を容易にする。

4.2 センサデータ解析支援システム RSV

4.1 節で提案した人間行動分析フレームワークにおいて、以下は手作業で行うと莫大な労力を要する。分析手順 1 において、膨大なセンサデータから解析に用いる多数の標本のセンサデータのみを抜き出すこと。分析手順 2 において、集めた標本のセンサデータを可視化すること。分析手順 4 において、データの傾向が見られた部分に対応するビデオデータを閲覧すること。

これらの作業を自動で行うシステムが存在すれば、人間行動分析を速やかに行える。そのようなシステムが満たすべき条件は次のようにまとめられる。

- C_1 = 膨大なセンサデータから必要なデータを抜き出せる (分析手順 1 の効率化)。
- C_2 = 多数の標本についてセンサデータを可視化できる (分析手順 2 の効率化)。
- C_3 = センサデータに対応するビデオデータを閲覧できる (分析手順 4 の効率化)。

そこで、本研究の課題を C_1, C_2, C_3 を満たすシステム RSV (Robot Sensor Viewer) の構築であると定式化する。 C_1 を満たすためには、分析に用いる標本を選択すること、その標本のセンサデータとビデオデータを検索するためのデータ (標本データと呼ぶ) を抜き出すことが求められる。 C_2 を満たすためには、 C_1 で抜き出した標本データを利用してセンサデータを検索し、検索されたセンサデータをセンサの種類に応じて可視化を行うことが求められる。 C_3 を満たすためには、 C_1 で抜き出した標本データからビデオデータ検索を行い、ビデオを再生することが求められる。

C_1 を満たすために、データ選択機能を RSV で実現する。 C_2 を満たすために距離センサグラフ、タッチセンサグラフを RSV で実現する。距離センサグラフ、タッチセンサグラフをまとめてグラフ表示機能と

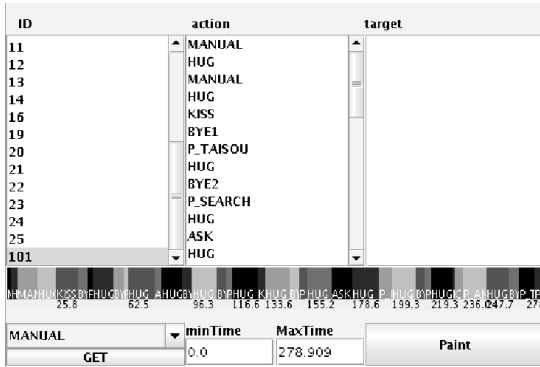


図 2 データ選択機能

Fig. 2 Data selection function.

呼ぶ。C₃ を満たすためにカメラ画像表示機能というカメラデータを動画として再生する機能を RSV で実現する。

RSV はこれらの機能を実現するために、センサデータである距離センサデータ、タッチセンサデータ、カメラデータのほかに、時刻（ミリ秒単位）を取得し検索のためのラベルを付与する。そして、データの取得開始からデータの取得終了までに得たデータすべてに同一の ID を割り振る。この ID をセッション ID と呼ぶ。ロボットと人間のインタラクション分析には、被験者ごとに分析する場合¹⁾ や、実験日ごとに分析する場合²⁾ がある。そこで、ユーザが被験者ごとや実験日ごとに個別のセッション ID を割り振ることで、データの検索を容易にする。

4.3 データ選択機能

データ選択機能は解析したい標本の選択を解析者に提供することで、膨大なセンサデータから特定のデータを抜き出す機能である（図 2）。データ選択機能にセッション ID 指定、ラベル選択、同名ラベル一括選択を提供することで、RSV は C₁ を満たす。各機能の詳細は以下に述べる。

データ選択を R-support で用いていた時刻指定で行うと、解析者の労力は大きい。その労力を軽減すべく、RSV は標本を提示し、解析者にそれを選択させることで、時刻指定を行わずにデータを抜き出すことを可能にする。

RSV における標本の扱い方を、RSV データ形式のサンプル（表 1）を例に述べる。表 1 ではセッション ID が 1 のときに、時刻 1, 2, 5, 6 で OHAYOU を実行している。これは 1 回の OHAYOU が途中で AKUSHU に中断されたのではなく、時刻 1, 2 と時刻 5, 6 で 2 回の OHAYOU が実行されたものと考えられる。そのため、ロボットが OHAYOU を実行

表 1 RSV データ形式

Table 1 RSV data format.

セッション ID	時刻	ラベル	センサデータ
1	1	OHAYOU	100,200,...
1	2	OHAYOU	100,200,...
1	3	AKUSHU	100,200,...
1	4	AKUSHU	100,200,...
1	5	OHAYOU	100,200,...
1	6	OHAYOU	100,200,...
2	1	OHAYOU	100,200,...

したときの人間行動を分析する際には、時刻 1, 2 の OHAYOU と時刻 5, 6 の OHAYOU はそれぞれ独立した標本として扱われる。

データ選択機能は C₁ を満たすために、セッション ID を指定した後に、そのセッション ID 内に存在するラベルの選択により、グラフ表示機能でセンサデータ検索を行う際に必要な標本データを抜き出す機能を提供する。標本データの具体的な内容は 5.2 節で述べる。またデータ選択機能は、セッション ID 内に同名ラベル標本が複数存在するときに一括で抜き出す機能も持つ。データ選択機能はこれらの機能を提供するためにセッション ID 指定、ラベル選択、同名ラベル一括選択を提供する。

データ選択機能はセッション ID を指定してから、そのセッション ID 内に存在するラベルを選択させる。そのため、たとえば被験者 1 の OHAYOU を分析するなど、特定被験者の特定ラベルの分析に優れている。

解析者がセッション ID を選択すると、セッション ID 指定は、そのセッション ID 内に存在するラベルを時系列に提示し、そのセッション ID 内に存在するラベルの種類を提示する。たとえば、表 1 のデータでセッション ID に 1 を指定した場合、OHAYOU, AKUSHU, OHAYOU と時系列提示し、OHAYOU, AKUSHU と種類提示する。ラベル選択は時系列に提示されたラベルから解析者が選択したラベルを抜き出し標本とする。同名ラベル一括選択は同一セッション ID 内における同名ラベルを一括で選択する。たとえば、表 1 でセッション ID に 1 を指定して OHAYOU を選択する場合は、時刻 1 ~ 2 の OHAYOU と時刻 5 ~ 6 の OHAYOU の 2 つの標本が抜き出される。RSV はこれらの機能で膨大なデータから必要なデータを 2 ステップで抜き出すことを実現する。一方、従来手法である R-support は絶対時刻指定をすることで標本を抜き出していた。RSV は数値入力をせずに標本を抜き出せるため、RSV は短時間で必要なデータを抜き出せる。

これ以外にも、データ選択機能は解析支援のために、

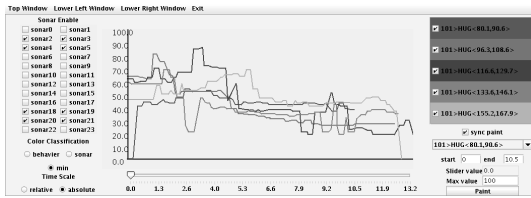


図 3 距離センサグラフ

Fig. 3 Sonar sensor graph.



図 5 カメラ画像表示

Fig. 5 Camera display function.

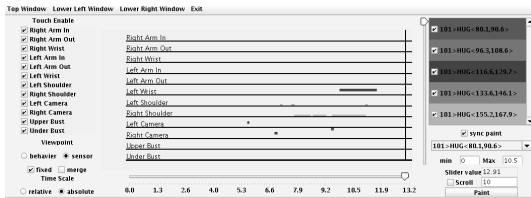


図 4 タッチセンサグラフ

Fig. 4 Touch sensor graph.

ラベル選択解除、ラベル遷移グラフ、そしてラベル遷移グラフの描画範囲指定を提供する。ラベル選択解除は、抜き出したラベルを取り消す機能である。ラベル遷移グラフは、図 2 の帯状のグラフであり、選択したセッション ID 内で行われたラベル遷移を可視化する機能である。遷移グラフの描画範囲指定は、ラベル遷移グラフの一部分を拡大表示する機能である。

4.4 グラフ表示機能

グラフ表示機能は、取得したセンサデータを可視化する機能であり、距離センサグラフとタッチセンサグラフがある。これら 2 つの機能はデータ選択機能で選択された多数の標本を一括して可視化する機能であり、これらを提供することで RSV は C_2 を満たす。

センサは一定周期ごとにデータを取得するため、センサデータは時系列データとなる。そこで、距離センサグラフとタッチセンサグラフは各センサデータの時系列の変化を可視化する。

距離センサデータは数値データであるため、数値データの時系列変化を観察できる折れ線グラフを用いて可視化する(図 3)。タッチセンサデータは 2 値データであるため、距離センサと同様に折れ線グラフを重ねて表示すると、グラフが複雑化する。そのため、タッチセンサグラフは帯グラフを並列表示する(図 4)。

タッチセンサグラフはグラフを並列表示するため、標本すべてを一括して可視化できない。そこで、RSV はグラフをスクロールする機能を提供することで、すべての標本について、タッチセンサデータを可視化する。

これ以外にも、グラフ表示機能は解析を支援するた

めに任意グラフの非表示、タイムライン、時間軸変更を提供する。任意グラフの非表示は、任意の標本と任意のセンサを無視した可視化をする機能である。タイムラインは、センサグラフに黒い線を表示し、グラフ上の時間を知る機能である。時間軸変更は、時間軸を絶対時間にするか、標本ごとの相対時間にするかを設定する機能である。

距離センサグラフは、さらにビュー指定、縦軸変更を提供する。ビュー指定は、配色数を減らすために、同一標本の距離センサを同色で塗るビュー、標本にかかわらず同一距離センサを同色で塗るビュー、そして距離センサの最小値を描画するビューの 3 種類から 1 つを選択する。縦軸変更は、縦軸の最大値を設定する。

タッチセンサグラフは、さらにビュー指定、ビューオプションを提供する。ビュー指定では、各標本でどのタッチセンサがいつ押されたかを可視化するビューと、各タッチセンサでどの標本でいつ押されたかを可視化するビューの 2 種類を提供する。標本やタッチセンサの数が多いときは、各系列に与えられる縦幅が小さくなり、タッチセンサの反応が見にくくなる。そのため、ビューオプションは標本やタッチセンサの種類に関係なく、反応の有無を検出するグラフを提供する。

4.5 カメラ画像表示機能

解析者は標本とカメラの種類を選択することでカメラ画像を表示できる(図 5)。標本とカメラの種類の選択はカメラ選択機能で行う。

Robovie はビデオ録画機能を有しない。そこで、カメラから静止画像のみを取得し、RSV はカメラ画像表示機能でカメラデータをアニメーションで切り替えて動画として再生する機能を提供する。そして、動画の再生に同期して距離センサグラフやタッチセンサグラフのタイムラインを動かす。これにより、RSV はセンサデータとカメラデータを対応付けて解析可能にする。したがって、RSV は C_3 を満たす。

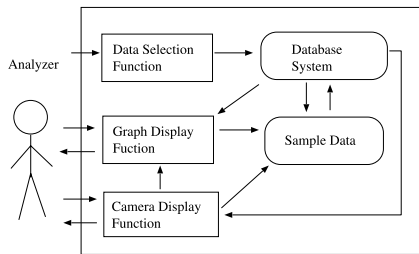


図 6 RSV の構成
Fig. 6 Structure of RSV.

5. RSV の設計と実装

5.1 全体の構成

RSV はデータ選択機能，グラフ表示機能，カメラ画像表示機能から構成される．データ選択機能は，センサデータやカメラデータ検索に必要なデータ（標本データ）を抜き出す．グラフ表示機能は，標本データを利用してセンサデータを検索し，センサの種類に応じて可視化する．カメラ画像表示機能は，標本データを利用してカメラデータを検索し，カメラデータを表示する．

RSV の全体構成を図 6 に示す．センサデータの格納にはデータベースシステムを用い，選択された標本をメモリに保存する．図 6 中のデータベースシステムへ入る矢印はクエリを示し，データベースシステムから出る矢印は結果の受け取りを示している．

たとえば，(A) 解析者がデータからラベル OHAYOU を抜き出し，(B) ラベル OHAYOU のタッチセンサデータを可視化して，(C) その後カメラデータを見たとする．この場合，システムは次のように動く．

A-(1) データ選択機能は解析者にデータベースに存在するセッション ID をすべてを提示する．

A-(2) 解析者はデータ選択機能で解析するセッション ID を選択する．

A-(3) データ選択機能は解析者が選択したセッション ID に存在するラベルすべてを提示する．

A-(4) 解析者がデータ選択機能で解析するラベル OHAYOU を選択する．

A-(5) データ選択機能は，データベースシステムからラベル OHAYOU の標本データを抜き出し，メモリに保存する．

B-(1) 解析者がタッチセンサグラフ（タッチセンサグラフはグラフ表示機能の 1 つ）を選択し，タッチセンサデータの可視化を指示する．

B-(2) タッチセンサグラフは，標本データを用いて，データベースシステムにクエリを発行し，データベ-

ースシステムからグラフ描画に必要なデータを取り出す．

B-(3) タッチセンサグラフは，取り出したデータの可視化を行い，解析者に提示する．

C-(1) 解析者は可視化されたデータから，特徴的な部分を発見し，その部分についてカメラ画像表示機能にカメラデータの表示を指示する．

C-(2) カメラ画像表示機能は，標本データを用いて，データベースシステムにクエリを発行し，カメラデータを表示するために必要なデータを取り出す．

C-(3) カメラ画像表示機能は，取り出したカメラデータを表示し，解析者に提示する．

5.2 RSV の特徴

5.2.1 データベース構造

本研究では Robovie からセンサデータとカメラデータを取得する．Robovie の性能ではカメラデータをセンサデータと同周期で取得できないため，センサデータとカメラデータを別テーブルに保存する．センサデータを保存するテーブルをセンサデータテーブルと呼び，カメラデータを保存するテーブルをカメラデータテーブルと呼ぶ．センサデータテーブルには「セッション ID，時刻，ラベル，距離センサ 1，距離センサ 2，…，距離センサ 24，タッチセンサ 1，タッチセンサ 2，…，タッチセンサ 16」などの合計 44 属性がある．カメラデータテーブルには「セッション ID，時刻，右目カメラデータ，左目カメラデータ，全方位カメラデータ」などの合計 5 属性がある．両テーブルは，ともにセッション ID と時刻が主キーである．

RSV は多数の標本について多数のセンサデータを可視化する．そのため，センサデータテーブルを分割して多数のテーブルでセンサデータを管理し，JOIN 演算を行ってデータ検索を行うと実行速度が低下する．解析支援システムの動作が遅くなることは解析者の不満につながる．そこで，本研究ではシンプルなテーブル構造を採用する．

5.2.2 アーキテクチャ上の特徴

関連研究である R-support のように時刻指定により標本を抜き出す場合，メモリに保存する標本のデータは開始時刻と終了時刻のみで十分である．しかし，RSV でラベル選択による標本抜き出しを実現するためには，セッション ID やセッション ID 内に存在するラベルをメモリに保存する必要がある．

そこで，RSV はセッション ID の種類を保存するセッション ID リスト，セッション ID 内に存在するラベルとその遷移時刻を保存するラベルリスト，抜き出した標本情報を保存する標本リストを持ち，それぞれメモリに保存する．標本リスト要素は「セッション

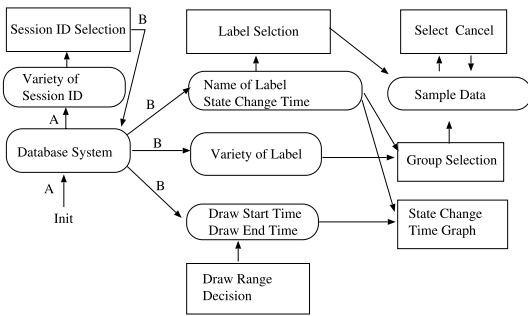


図 7 データ選択機能の構成

Fig. 7 Structure of data selection function.

ID, ラベル名, 標本開始時刻, 標本終了時刻」であり, これを標本データとする. RSV はこれら 3 つのリストを用いて, データ選択機能で選択された標本の標本データを標本リストに保存する.

RSV は標本データのセッション ID, 標本開始時刻, 標本終了時刻を利用してセンサデータテーブルとカメラデータテーブルからデータを検索する. 検索されたデータは膨大となるため, メモリに保存せず, グラフ描画やカメラ表示が必要となるたびにデータを検索する.

RSV はセッション ID リスト, ラベルリスト, 標本リストを持つことで, 標本選択を可能にする. そして, 標本リストに保存された標本データからデータベースシステムへクエリを発行し, センサデータとカメラデータを取得・提示する.

5.2.3 インタフェース上の特徴

情報や機能が散在すると, 解析にかかる労力が増えてしまう. そこで, RSV は解析者が求める情報や機能を 1 つの画面に収めるインタフェースとなっている. これを実現するために, RSV は 1 つのコンポーネントに多種類の情報を持たせている. たとえば, 図 3 の右上では, 選択された標本と標本開始時刻と標本終了時刻を表示するとともに, 色を付けることでグラフの凡例の役目も持たせている. さらに, チェックボックスを利用することで特定標本を非表示にする機能を持たせている.

5.3 データ選択機能

データ選択機能は, セッション ID 指定, ラベル選択, ラベル選択の解除, 同名ラベル一括選択, ラベルの遷移グラフ, そして遷移グラフの範囲指定から構成される. 各機能とデータの流れを図 7 に示す. 図中の A, B はクエリ発行と結果受け取りの対応を示している.

RSV は起動時にデータベースシステムに存在する

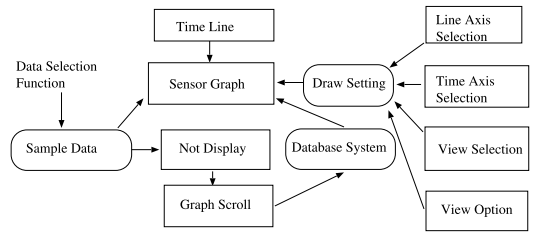


図 8 グラフ表示機能の構成

Fig. 8 Structure of graph display function.

セッション ID を調べ (図 7 クエリ A), メモリに保存しておく. そして, データ選択機能を表示する際にセッション ID を解析者に示す.

セッション ID 指定で解析者がセッション ID を選択すると, 選択したセッション ID 内に存在するラベルを調べるため, データベースシステムへクエリを発行し (図 7 クエリ B), 時刻とその時刻のラベルを受け取る. そして, 時刻とラベルから以下のデータを求めメモリに保存する: 存在するラベルとその遷移時刻, 存在するラベルの種類, 遷移グラフの描画開始・終了時刻. さらに, ラベルの時系列提示と種類提示を行う.

ラベル選択で解析者が時系列提示されたラベルを選択すると, 標本データがメモリに保存される. そして, 選択したラベルでグラフが描画される.

同名ラベル一括選択で解析者がラベルの種類を指定すると, ラベルと遷移時刻のデータから, 選択する標本を特定し, 標本データをメモリに保存する.

ラベル選択解除では, 選択された標本データを破棄する. ラベル遷移グラフでは, 遷移グラフの描画開始・終了時刻, ラベルとその遷移時刻から, 選択したセッション ID でのラベル遷移グラフを描画する. 遷移グラフの範囲指定では, ラベル遷移グラフの描画開始・終了時刻を変更する.

5.4 グラフ表示機能

グラフ表示機能は距離センサグラフとタッチセンサグラフの 2 種類を持つ. グラフ表示機能の各機能とデータの流れを図 8 に示す. 距離センサグラフは, 距離センサグラフ表示, ビュー指定, 任意グラフの非表示, タイムライン, 縦軸変更, 時間軸変更から構成される. タッチセンサグラフはタッチセンサグラフ表示, グラフスクロール, ビュー指定, 任意グラフの非表示, タイムライン, 時間軸変更, ビューオプションから構成される. ここで, ビュー指定, 縦軸変更, 時間軸変更, ビューオプションによる設定を描画設定と呼ぶ.

センサグラフの描画は以下のようにしてなされる. 標本データ, 任意グラフの非表示, グラフスクロール (タッチセンサグラフのみ) から描画する標本を決定

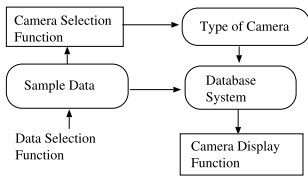


図 9 カメラ画像表示の構成
Fig.9 Structure of camera display function.

し、データベースシステムからその標本の、時間とセンサデータを受け取る。そして、時間、センサデータ、描画設定、標本データ（タッチセンサグラフのみ）を組み合わせるセンサグラフを描画する。

タッチセンサグラフのグラフスクロールは、選択した標本中で描画する標本を決定する。描画する標本が変更されるたびにタッチセンサグラフを描画する。

ビュー指定は、ビュー種類を解析者に設定させる。任意グラフの非表示では、解析者が指定した標本とセンサを非表示に設定する。タイムラインは、解析者が指定した時刻からセンサグラフ上の位置を求め、タイムラインを表示する。縦軸変更は、縦軸の最大値を解析者の指定値に設定する。時間軸変更は、時間軸を解析者の指定時間軸に設定する。ビューオプションは、解析者の指定オプションに設定する。

5.5 カメラ画像表示機能

カメラ画像表示機能は、カメラ選択機能とカメラ画像表示から構成される。各機能とデータの流れを図 9 に示す。

カメラ選択機能は、標本とカメラ種類を解析者に提示する。解析者が標本とカメラ種類を指定すると、カメラ画像表示が行われる。

カメラ画像表示は、解析者の指定時刻に対応するカメラデータをデータベースシステムから検索し表示する。動画再生時は、動画再生スレッドを作成し、カメラデータをアニメーション表示させる。このとき、アニメーションに同期して、距離センサグラフやタッチセンサグラフのタイムラインを動かす。動画停止時には、作成したスレッドを停止して、アニメーションを止める。

5.6 実装

RSV の実装には JDK 1.5.0 を使用した。データ格納には PostgreSQL を使用した。ソースコードは 3,000 行程度を要した。

6. 実行例

4.1 節で本研究が提案する 5 手順の分析フレームワークを述べた。本章では、RSV を用いることで提

表 2 行動モジュールの種類

Table 2 Behavior modules.

INAINAI	いないいないばあをする
OHAYOU	おっはーと発話
SANZYOU	Robovie 参上と発話
ASOBO	触ってねと発話
ZYAN	じゃんけんを求める
AKUSHU	握手を求める
KISS	キスを求める
ATTIMUITE	あっちむいてをを求める
HUG	抱擁を求める
TAISOU	体操をする
USONAKI	うそ泣きをする
KOCHO	こちょこちょをする
TSUMARANAI	つまらないと発話
TIRED	つかれたと発話
SLEEP	ねむたいよと発話

表 3 各行動モジュールの認識方法

Table 3 Recognition methods of behavior modules.

モジュール名	行動検出方法
INAINAI	全方位カメラによる正面方向の移動物体
OHAYOU	全方位カメラによる正面方向の移動物体
SANZYOU	全方位カメラによる正面方向の移動物体
ASOBO	両腕と胴体のタッチセンサ反応
ZYAN	全方位カメラによる人間の手認識
AKUSHU	右腕、右手首のタッチセンサ反応
KISS	全方位カメラによる正面方向の移動物体
ATTIMUITE	全方位カメラから人間の向いた方向を認識
HUG	距離センサ値の減少、肩のタッチセンサ反応
TAISOU	全方位カメラによる正面方向の移動物体
USONAKI	タッチセンサの反応
KOCHO	腕のタッチセンサ反応
TSUMARANAI	なし
TIRED	全方位カメラによる正面方向の移動物体
SLEEP	全方位カメラによる正面方向の移動物体

案したフレームワークで人間行動を分析し、一般的な人間行動を検出できることを検証実験を通して示す。

6.1 実験内容

Robovie の行動プログラムは多数の行動モジュール¹²⁾ から構成されるという特徴を持つ。行動モジュールは握手、じゃんけんなどの単一の行動に対応するモジュールであり、単一の行動を実行する際の発話、モーション、人間行動を検出する条件が記述されている。検証実験では表 2 に示す 15 種類の行動モジュールについて人間行動を分析する。また、これらの行動モジュールが認識に用いる方法を表 3 に示す。そして、各行動モジュールについてセンサデータ解析を行うために、センサデータを抜き出すためのラベルとして行動モジュールの名前を採用する。

検証実験では、6 人の被験者に 15 分程度 Robovie の相手をさせた。この間、表 2 に示す 15 種類の行動モジュールを Robovie に 5 回ずつ実行させ、各モジュールにつき 30 個の標本を採集した。セッション ID は 1

人に1つを割り当てた。割り当てたセッションIDは127から132である。ここで、被験者にはRobovieに友好的に対応すること、どうすればRobovieが反応してくれるかを考えて対応することを要求した。この条件を提示することで、人間がRobovieに友好的に接するとき、どのような行動をとるかをRSVを用いて分析する。分析の際には、距離センサとタッチセンサを用いて人間行動の検出を試みる。

6.2 分析の手順

本実行例では次の手順で人間行動分析を行う。

- 分析手順 1 30個の標本を抜き出す。
- 分析手順 2 センサデータを可視化する。
- 分析手順 3 可視化データから人間行動を発見する。
- 分析手順 4 発見した人間行動部分のカメラデータを表示。
- 分析手順 5 カメラデータから人間行動を確認する。

この分析手順1から分析手順5までを各モジュールについて行う。この中で、分析手順1, 2, 4はRSVが支援することで効率的に実現される。

次節よりRSVを用いた分析について述べる。6.3節で30個の標本抜き出しを述べる(分析手順1)。6.4節でセンサデータを可視化し人間行動を発見することを述べる。ここでは15のモジュールの中から特徴的な5つのモジュール(INAIINAI, ASOBO, AKUSHU, HUG, KOCHO)を取り上げる(分析手順2)。6.5節では5つのモジュールで発見できた人間行動をまとめる(分析手順3)。6.6節でカメラデータの表示と人間行動確認を述べる(分析手順4, 5)。

6.3 標本の選択

15のモジュールを個別分析する際に、標本選択をそれぞれ行うが、標本選択の方法は15のモジュールすべてについて同じ方法を用いる。そこで、本節ではINAIINAIモジュールの標本選択を述べる。

6人の被験者のセッションIDは127から132である。また、1人につき5回ずつINAIINAIを実行しているため、1つのセッションID内に5つの標本が存在する。具体的な抜き出し方法は次のとおりである。

- セッションID127を選択し、同名ラベル一括選択を用いてINAIINAIを抜き出す。
- 同様の操作をセッションID128, 129, 130, 131, 132で行う。

以上の手順で6つのセッションIDからそれぞれ5つ、計30個の標本を抜き出せる。標本を抜き出したときのデータ選択機能の表示を図10に示す。図10右のリストに抜き出された標本が表示されていることが分かる。また、30個の標本の抜き出しにかかる時

ID	action	target
117	MANUAL	127>P_INAIINAI<17.6,31
118	P_OHA	127>P_INAIINAI<211.0,2
120	P_SANZYOU	127>P_INAIINAI<346.6,3
121	P_INAIINAI	127>P_INAIINAI<442.3,4
122	ASOBO	127>P_INAIINAI<641.3,6
125	ZYAN	128>P_INAIINAI<15.9,30
126	AKUSHU	128>P_INAIINAI<201.6,2
127	KISS	128>P_INAIINAI<356.3,3
128	ATTIMUITE	128>P_INAIINAI<452.0,4
129	HUG	128>P_INAIINAI<642.2,6
130	TAISOU	129>P_INAIINAI<16.9,32
131	USO_NAKI	129>P_INAIINAI<212.8,2
132	KOCHO_KOCHO	

図10 INAIINAIの標本選択

Fig. 10 Sample selection of INAIINAI.

間を計測したところ、約25秒であった。

6.4 RSVによるセンサデータの分析

6.4.1 INAIINAIモジュールの分析

INAIINAIモジュールとは、Robovieが「いないいない」と発話し、全方位カメラの画像から正面方向に移動物体があるかどうかを求めて人間の存在を検出するモジュールである。

距離センサデータから人間行動を分析するために、距離センサグラフを表示する。ここで、人間はRobovieの正面でインタラクションしていると仮定して、前方の距離センサ以外を非表示にし、さらに最も近付いたと認識した距離センサの値のみを見るために最小値表示にする。こうしてできたグラフが図11である。このグラフより、初めは40~60cmほどの値が出ており、途中から20cm程度まで小さくなるのが分かる。ほぼ全標本において同じ変化が観察されたため、人間行動でない要因がデータ変化に影響したと考えられる。そこで、RobovieにINAIINAIモジュールを実行させたところ、値が小さくなった原因は人間行動によるものではなく、動かした自分の腕を距離センサが感知したためであることが分かった。それゆえ、人間行動の認識に距離センサを使えないことが分かった。

タッチセンサデータから人間行動を分析するために、タッチセンサグラフを表示する。すべてのタッチセンサをグラフ化したものが図12である。ここで右腕内側、右腕外側、左腕内側、左腕外側、左手首のタッチセンサで比較的反応が見られたため、それぞれのセンサについてグラフを表示させる。右腕内側と右腕外側のグラフを図13に、左腕内側と左腕外側のグラフを図14に、左手首のグラフを図15に示す。これらのグラフより、人間は右腕内側、右腕外側、右手首、左手首のタッチセンサに触れることが分かる。

以上より、INAIINAIモジュールにおいて、距離センサは自分の腕を感知するため人間行動の検出には使

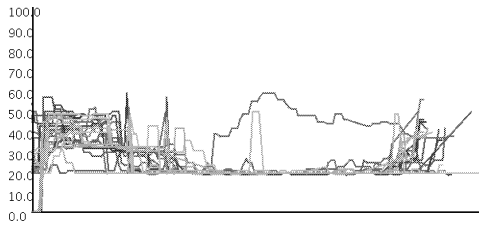


図 11 INAIINAI の距離センサグラフ
Fig. 11 Sonar sensor graph of INAIINAI.



図 15 INAIINAI の左手首センサグラフ
Fig. 15 Left wrist sensor graph of INAIINAI.

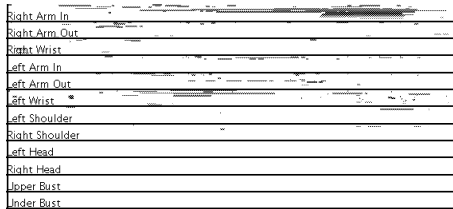


図 12 INAIINAI のタッチセンサグラフ
Fig. 12 Touch sensor graph of INAIINAI.

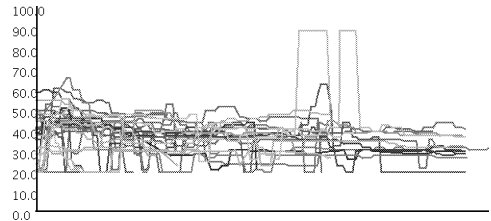


図 16 ASOBO の距離センサグラフ
Fig. 16 Sonar sensor graph of ASOBO.

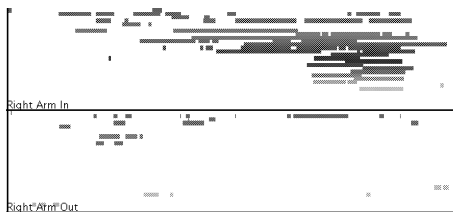


図 13 INAIINAI の右腕センサグラフ
Fig. 13 Right arm sensor graph of INAIINAI.

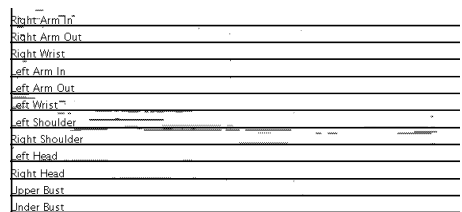


図 17 ASOBO のタッチセンサグラフ
Fig. 17 Touch sensor graph of ASOBO.



図 14 INAIINAI の左腕センサグラフ
Fig. 14 Left arm sensor graph of INAIINAI.

距離センサグラフ (図 16) より, 距離センサ値が 20 ~ 60 cm の間で変動することが分かる. よって, 対人距離は 20 ~ 60 cm であることが分かる. また, タッチセンサグラフ (図 17) より, 一般的に人間は左肩, 右肩のタッチセンサを触れること, 少数の人間は左頭, 右頭のタッチセンサを触れることが分かる.

以上より, ASOBO モジュールにおいて, 人間と Robovie の距離が 20 ~ 60 cm であること, 一般的に人間は左肩, 右肩のタッチセンサを触れること, 少数の人間は左頭, 右頭のタッチセンサを触れることが分かる. ASOBO モジュールは両腕と胴体のタッチセンサから人間行動認識を測ろうとしたが, 実際には人間は両腕と胴体を触れる傾向がないことが分かる. 一方, 人間は肩や頭のタッチセンサを触れることが分かったため, 肩と頭のタッチセンサ反応を人間行動認識に組み込むことで, 認識精度の向上を図れる.

用できないこと, 人間は右腕内側, 右腕外側, 右手首, 左手首のタッチセンサに触れることが分かる. INAIINAI モジュールは全方位カメラから移動物体があるかどうかの認識のみ行っているが, タッチセンサの反応を人間行動認識に組み込むことで, 認識精度の向上を図れる.

6.4.2 ASOBO モジュールの分析

ASOBO モジュールとは, Robovie が「触ってね」と発話し, 両腕と胴体のタッチセンサ反応から人間が Robovie に触れたことを認識するモジュールである.

6.4.3 AKUSHU モジュールの分析

AKUSHU モジュールとは, Robovie が「握手して」と発話し右手を差し出し, 右腕内側, 右腕外側, 右手

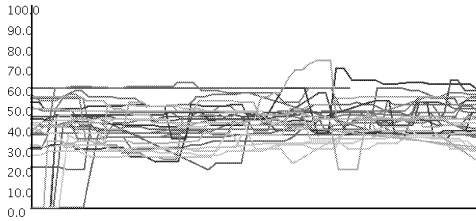


図 18 AKUSHU の距離センサグラフ
Fig. 18 Sonar sensor graph of AKUSHU.

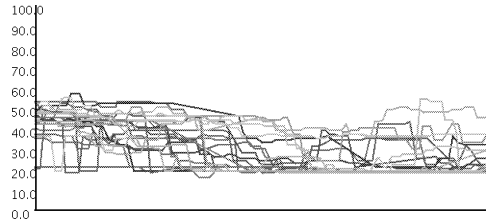


図 20 HUG の距離センサグラフ
Fig. 20 Sonar sensor graph of HUG.

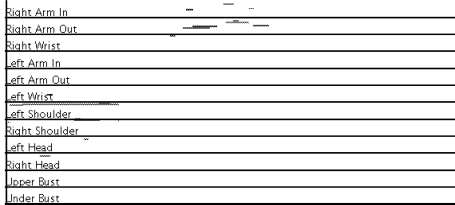


図 19 AKUSHU のタッチセンサグラフ
Fig. 19 Touch sensor graph of AKUSHU.



図 21 HUG のタッチセンサグラフ
Fig. 21 Touch sensor graph of HUG.

首のタッチセンサの反応から人間の握手を認識するモジュールである。

距離センサグラフ (図 18) より, 距離センサ値が 20~60 cm の間で変動することが分かる。よって, 対人距離は 20~60 cm であることが分かる。また, タッチセンサグラフ (図 19) より, 右腕や右手首のタッチセンサの反応はそれほど多くなく, 少数の人間は左肩や右肩を触れることが分かる。

以上より, AKUSHU モジュールにおいて, 人間と Robovie の距離が 20~60 cm であること, 人間はあまり右腕や右手首のタッチセンサに触らないこと, 少数の人間は左肩や右肩のタッチセンサを触れることが分かる。AKUSHU モジュールは右腕を出して握手しようとする働きかけるモジュールであるにもかかわらず, 右腕や右手首のタッチセンサの反応はあまりなかった。ゆえに, 人間が握手を返したことを認識するには何か別の手段が必要になると考えられる。また, 人間が肩のタッチセンサを触れる意図を究明できれば, 肩のタッチセンサ反応を人間行動認識に組み込むことで, 認識精度の向上を図れる。

6.4.4 HUG モジュールの分析

HUG モジュールとは, Robovie が「だっこして」と発話しながら腕を広げ, 距離センサにより人間が近付いたこと, もしくは人間が肩のタッチセンサを触ったことを認識し, 人間行動を認識できた場合に「好き」や「大好き」と発話するモジュールである。

距離センサグラフ (図 20) より, 途中から距離センサの値が 20~40 程度まで減少することが分かる。

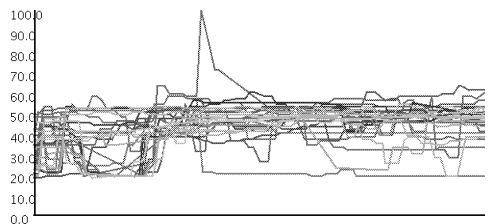


図 22 KOCHO の距離センサグラフ
Fig. 22 Sonar sensor graph of KOCHO.

よって, 人間が近付いたことが分かる。また, タッチセンサグラフ (図 21) より, 人間は左肩, 右肩のタッチセンサを触れることが分かる。

以上より, HUG モジュールにおいて, 人間が近付いたことにより距離センサ値が 20~40 cm になること, 人間は左肩や右肩のタッチセンサを触れることが分かる。HUG モジュールは人間行動を距離センサの変化と左肩と右肩のタッチセンサの変化で認識する設計であったため, HUG モジュールの設計は実際の人間行動に即したものであるといえる。

6.4.5 KOCHO モジュールの分析

KOCHO モジュールとは, Robovie が手を前に出し, こちょこちょとくすぐり, 手先に人間が触れるかどうかを検出するモジュールである。

距離センサグラフ (図 22) より, 途中から距離センサ値が 40~60 cm 程度まで大きくなる事が分かる。よって, 対人距離が 40~60 cm 程度であることが分かる。また, タッチセンサグラフ (図 23) より, 人間は右手首, 左手首, 右腕内側, 右腕外側, 左腕内側, 左

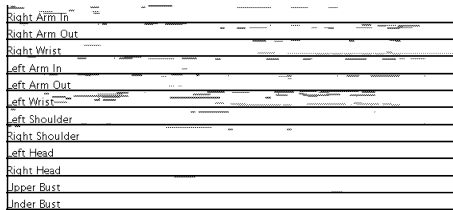


図 23 KOCHO のタッチセンサグラフ

Fig. 23 Touch sensor graph of KOCHO.

腕外側のタッチセンサに触れることが分かる。

以上より、KOCHO モジュールにおいて、人間は Robovie から 40~60cm 程度の距離まで離れること、人間は右手首、左手首、右腕内側、右腕外側、左腕内側、左腕外側のタッチセンサに触れることが分かる。KOCHO モジュールは右手首と左手首のタッチセンサの反応で人間行動を検出しているが、実際の間人も右手首と左手首のタッチセンサに触れる。それゆえ、KOCHO モジュールの人間行動認識は実際の間人間の行動に即したものであると考えられる。

6.5 分析により発見できた人間行動

6.4 節で述べた 5 つのモジュールの分析結果を表 4 に示す。表 4 の [発見] とは新しい人間行動の発見を示す。[誤解] とはモジュール設計者が予測した行動を人間がとらなかったことを示す。[追認] とはモジュール設計者が予測した行動を人間がとったことを示す。

INAIINAI モジュールではタッチセンサ反応を新たに発見できた。ASOBO モジュールでは開発者が予測した両腕と胴体のタッチセンサ反応はないことを発見するとともに、両肩、左頭、右頭のタッチセンサ反応が新たに発見できた。AKUSHU モジュールでは開発者が予測した右腕と右手首のタッチセンサ反応が少ないことを発見するとともに、両肩のタッチセンサ反応が若干あることを発見できた。HUG モジュールでは、開発者が予測した人間が近付くこと、および左肩、右肩のタッチセンサに人間が触ることが正しいことが分かった。KOCHO モジュールでは、開発者が予測した右手首と左手首に人間が触れることが分かるとともに、人間が遠ざかることを発見した。

6.6 カメラデータによる人間行動確認

カメラデータによる人間行動確認の中から、INAIINAI モジュールの距離センサが認識には使えない点と、HUG モジュールで人間が近付いた点と、KOCHO モジュールで人間が遠ざかった点について述べる。

ある被験者の INAIINAI モジュールで距離センサの値が下がっていた部分のカメラ画像を図 24 に、HUG モジュールで距離センサの値が下がっていた部分の

図 24 INAIINAI カメラ画像
Fig. 24 Photo of INAIINAI.図 25 HUG カメラ画像
Fig. 25 Photo of HUG.図 26 KOCHO カメラ画像
Fig. 26 Photo of KOCHO.図 27 AKUSHU カメラ画像
Fig. 27 Photo of AKUSHU.

メラ画像を図 25 に、KOCHO モジュールで距離センサの値が上がっていた部分のカメラ画像を図 26 に示す。

図 24、図 25、図 26 より、HUG モジュールにおいて人間が最も近付いていることが分かる。一方、図 11 と図 20 を見比べると、HUG モジュールより INAIINAI モジュールの方が値が小さい。よって、INAIINAI モジュールで距離センサの値が小さくなったのは人間行動によるものではなく、別の要因があると考えられる。つまり、INAIINAI モジュールでは距離センサは認識に使えないことが確認できた。また、HUG モジュールで人間が近付いたこと、KOCHO モジュールで人間が遠ざかったことも確認できた。

次に、AKUSHU モジュールで人間が肩を触った意図について述べる。ある被験者が AKUSHU モジュールで肩のタッチセンサを触ったときのカメラ画像を図 27 に示す。図 27 を見ると、顔を近付けていることから、Robovie に対して興味を持っていることは分かる。しかし、肩を触った意図まではカメラ画像で確認できなかった。

7. 議 論

本章では、人間とロボットのコミュニケーションから得たセンサデータを、RSV で分析する利点を述べる。

7.1 標本の選択

RSV はデータ選択機能を用いて、各モジュールで

表 4 RSV による人間行動分析のまとめ
Table 4 Summary of human behavior analyses by RSV.

モジュール名	新たな発見	詳細
INAIINAI		[発見] 距離センサは人間行動認識に使えない [発見] 人間は右腕内側, 右腕外側, 右手首, 左手首のタッチセンサに触れる傾向がある
ASOBO		[誤解] 人間行動認識に用いられていた両腕と胴体のタッチセンサに人間は触れない [発見] 人間は左肩, 右肩のタッチセンサに触れる [発見] 左頭, 右頭のタッチセンサに触れる人も少数いる
AKUSHU		[誤解] 人間行動認識に用いられていた右腕と右手首のタッチセンサに人間はあまり触れない [発見] 左肩, 右肩のタッチセンサに触れる人が少数いる
HUG	x	[追認] 人間が近付くという条件は正しい [追認] 左肩, 右肩のタッチセンサに人間が触れるという条件は正しい
KOCHO		[追認] 人間行動認識に用いられていた右手首と左手首のタッチセンサに多くの人間は触れる [発見] 人間は遠ざかる

30 個の標本を抜き出せた。それゆえ、RSV はセンサデータの中から解析する標本を多数抜き出すという要件を満たしているといえる。

7.2 センサデータからの人間行動発見

HUG モジュールと KOCHO モジュールでは、センサデータを可視化することで、人間がとる行動とプログラムの人間行動認識手法が対応していることが分かった。たとえば、HUG モジュールは距離センサによる人間の接近と両肩のタッチセンサで人間行動を認識するよう設計されているが、実際に人間が近付くことおよび両肩のタッチセンサに触れることが分かった。すなわち、RSV は行動プログラムが実際の人間行動に即していることを確認可能にする。

ASOBO モジュールと AKUSHU モジュールでは、センサデータを可視化することで、行動プログラム設計者の予測行動を人間がとらないことが分かった。たとえば、ASOBO モジュールは両腕と胴体のタッチセンサで人間行動を認識するよう設計されているが、被験者は両腕と胴体のタッチセンサに触れなかった。すなわち、RSV は誤った人間行動の認識手法を発見できる。

Robovie の行動モジュールの多くは全方位カメラのみで人間の行動を認識していたが、タッチセンサの有用性が分かった。たとえば、INAIINAI モジュールにおいて人間は右腕内側、右腕外側、右手首、左手首のタッチセンサに触れる傾向が見られた。これらを新たに人間行動の認識手法として行動プログラムに組み込むことで、より高い精度で認識可能になると考えられる。

以上より、RSV を用いて人間行動を分析することで、行動プログラムが実際の人間行動に即していることの確認、誤った人間行動の認識手法の発見、新たな人間行動の認識手法の発見が可能になるといえる。

7.3 カメラデータによる人間行動確認

実行例で、HUG モジュールにおいて人間が近付いたこと、KOCHO モジュールで人間が遠ざかったこと、INAIINAI モジュールで距離センサが人間行動認識に使用できないことが確認できた。一方、AKUSHU モジュールで人間が肩のタッチセンサに触れる意図を確認することができなかった。よって、RSV はある程度の人間行動確認能力を備えているが、すべての人間行動確認は行えないといえる。

人間行動確認ができなかった理由として、Robovie の目のカメラを採用したことが考えられる。Robovie の目からでは人間の全身をとらえられないため、人間行動の確認範囲が狭まったと考えられる。この問題は、人間の全身をとらえられる外部カメラの利用により解決できる。RSV で外部カメラを利用可能とするには、カメラデータテーブルに外部カメラの属性を加え、Robovie のセッション ID と時刻を同期させてカメラデータを取得すればよい。

7.4 関連研究との機能・性能面の比較

7.4.1 標本選択

RSV が 30 個の標本抜き出しに要した時間は約 25 秒であった。ビデオで標本抜き出しを行う場合、早送りや巻戻し、もしくはビデオの再生開始時刻の指定を行うため数十分を要する。

また、R-support を用いて標本の抜き出しを行う場合、それぞれの標本の開始時刻と終了時刻を指定しなければならない。そのため、R-support で標本の抜き出しを行うと 10 数分かかる。

すなわち、RSV は従来のシステムにくらべ、標本選択の労力を減らす。

7.4.2 人間行動の発見

RSV はセンサデータの可視化により人間行動を発見可能にした。RSV は 30 個の標本を一括して可視化するため、標本間の比較が容易となり、人間行動の発

見が容易になる。

ビデオを用いて人間行動を発見する場合、ビデオを見ながら特徴的な行動を発見し、他の人間も発見した特徴的な行動をとるかどうかをビデオを見て確認しなければならない。そのため、莫大な時間と労力がかかり、見落としも発生してしまう。

R-support を用いて人間行動を発見する場合、多数の標本を一括して可視化できないため、30 個の標本を比較しながら分析を行うことは不可能である。

すなわち、RSV は多数の標本の比較分析に優れ、従来のシステムにくらべ、適切な人間行動を発見できる。

7.4.3 ま と め

以上をまとめると、RSV は従来システム R-support に比べて、標本選択にかかる労力を削減し、多数の標本の比較分析に優れ、適切な人間行動を発見できるといえる。

7.5 関連研究とのシステム技術面の比較

7.5.1 データ管理

RSV はセンサデータ管理にデータベースシステムを利用している。

R-support ではファイルでセンサデータを管理していたが、センサデータは取得し続けると膨大な量となるため、ファイルでのデータ管理には限界がある。また、グラフ描画の際に毎回ファイルを読み込まなければならないため、センサデータが膨大な量になるとグラフ描画に遅延が発生すると考えられる。

RSV はデータベースシステムでのデータ管理を実現するために、PostgreSQL を利用した。データベースシステムを利用することで、センサデータが膨大になった場合への対処を行った。今回は速度向上の試みを行わなかったが、6 章での実行例程度のデータ規模では、グラフ描画の遅延は見られなかった。しかし、今後データ量が増えれば、キャッシング技術などによるデータ検索速度の向上が求められるだろう。

7.5.2 標本選択

RSV はセッション ID 選択とラベル選択の 2 手順を用いて標本選択を行い、標本の描画範囲を標本データとして保存する。

R-support では時刻指定を用いて描画範囲を指定していたが、標本数が増えると、時刻指定の作業量は莫大になる。

RSV は標本選択による描画範囲指定を行うために、標本マネージャとしてデータ選択機能を作成した。標本マネージャはセッション ID リスト、ラベルリスト、標本リストを持ち、解析者の選択に応じてラベルリストと標本リストを更新する機能を持つ。

7.5.3 グラフ描画

RSV は複数標本の一括可視化を行う。

R-support は単一標本の可視化のみを行っていた。そのため、多数の標本を比較して人間行動を発見することは難しい。

RSV は折れ線グラフの多重化と帯グラフの並列化により、複数標本の可視化を実現した。また、多重化や並列化により、複雑さや画面に表示しきれないという問題が発生した。そこで、配色変更、距離センサの最小値表示、グラフのスクロールを可能とすることで、複雑さや画面に表示しきれない問題を解決した。

8. 結 論

本論文ではセンサデータを利用した人間行動分析フレームワークと分析を支援するソフトウェア RSV を提案し、提案の有用性を検証するために人間とロボットをコミュニケーションさせて得たセンサデータを RSV によって分析し人間行動を抽出した。RSV を用いることで多数の標本を容易に抜き出すことが可能になった。RSV を用いて分析することで、行動プログラムが実際の人間行動に即していることの確認、誤った人間行動の認識手法の発見、新たな人間行動の認識手法の発見が可能になった。そして、RSV はカメラ画像による人間行動確認ができる。本論文は、コミュニケーションロボットの行動プログラム開発において、RSV を用いた人間行動分析を行うことで、より実際の人間行動に即した行動プログラムを作成できると結論する。

参 考 文 献

- 1) Kanda, T., Ishiguro, H., Ono, T., Imai, M. and Nakatsu, R.: Development and Evaluation of an Interactive Humanoid Robot "Robovie", *IEEE International Conference on Robotics and Automation (ICRA 2002)*, pp.1848-1855 (2002).
- 2) Kanda, T., Hirano, T., Eaton, D. and Ishiguro, H.: A practical experiment with interactive humanoid robots in a human society, *IEEE International Conference on Humanoid Robots* (2003).
- 3) 佐藤英和: Robovie を用いた実験を支援するソフトウェア R-support の設計と実装, 修士論文, 慶應義塾大学 (2004).
- 4) 神田崇行, 石黒 浩, 今井倫太, 小野哲雄: 人-ロボット相互作用における身体動作の数値解析協調的動作の重要性, *情報処理学会誌*, Vol.44, No.11, pp.2699-2709 (2003).
- 5) 斎藤康彦: 多変量データ系列における規則性を発見するための可視化手法, *情報処理学会論文誌*:

データベース, Vol.40, No.SIG6(TOD3), pp.1-11 (1999).

- 6) 大東 誠, 田中 譲: データベース・レコードの仮想実体化フレームワーク, 情報処理学会論文誌: データベース, Vol.42, No.SIG1(TOD8), pp.80-91 (2001).
- 7) 堀田政二, 井上光平, 浦浜喜一: ファジークラスタリングによる視覚化と検索のグラフ構造データへの応用, 情報処理学会論文誌: データベース, Vol.42, No.SIG1(TOD8), pp.70-79 (2001).
- 8) 尾下真樹, 牧之内顕文: オブジェクト指向データベースの半自動可視化環境, 情報処理学会論文誌: データベース, Vol.42, No.SIG1(TOD8), pp.56-59 (2001).
- 9) Fujishiro, I., Furuhashi, R., Ichikawa, Y. and Takeshima, Y.: GADGET/IV: A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environment, *Proc. IEEE Symposium on Information Visualization*, pp.77-83 (2000).
- 10) Girod, L., Elson, J., Cerpa, A., Stathopoulos, T., Ramanathan, N. and Estrin, D.: EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks, *Proc. USENIX* (2004).
- 11) Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohier, E. and Estrin, D.: Sympathy for the Sensor Network Debugger, *Proc. Embedded networked sensor systems (Sensys)*, pp.255-267 (2005).
- 12) Ishiguro, H., Kanda, T., Kimoto, K. and Ishida, T.: A Robot Architecture Based on Situated Modules, *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems*, pp.1617-1623 (1999).

(平成 17 年 12 月 20 日受付)

(平成 18 年 4 月 25 日採録)

(担当編集委員 金子 邦彦)



中村 学 (学生会員)

2005 年慶應義塾大学理工学部情報工学科卒業。現在, 同大学大学院開放環境科学専攻前期博士課程在学中。センサネットワークの研究に従事。日本データベース学会学生会員。



川島 英之 (正会員)

2005 年慶應義塾大学大学院開放環境科学専攻後期博士課程修了。博士(工学)。現在, 慶應義塾大学理工学部情報工学科助手。データベースシステムの研究に従事。ACM 会員。



佐竹 聡 (学生会員)

2003 年慶應義塾大学理工学部情報工学科卒業。2005 年同大学大学院開放環境科学専攻前期博士課程修了。現在, 同大学院開放環境科学専攻後期博士課程在学中。ヒューマンロボットインタラクション, センサネットワークの研究・開発に従事。



今井 倫太 (正会員)

1992 年慶應義塾大学理工学部電気工学科卒業。1994 年同大学大学院計算機科学専攻修士課程修了。同年 NTT ヒューマンインターフェース研究所入社。1997 年 ATR 知能映像通信研究所へ出向。2002 年慶應義塾大学大学院開放環境科学専攻博士課程修了。現在, 慶應義塾大学理工学部情報工学科助教授, および, ATR 知能ロボティクス研究所客員研究員。ロボットとの対話, センサを用いた状況知覚に興味を持つ。電子情報通信学会, 人工知能学会, IEEE, ACM 等会員。博士(工学)。



安西祐一郎 (正会員)

1974 年慶應義塾大学大学院博士課程修了。1988 年より慶應義塾大学理工学部教授, 1994 年より理工学部長。2001 年より慶應義塾長。この間 1981~1982 年カーネギーメロン大学客員助教授。計算機科学, 認知情報処理過程の研究に従事。電子情報通信学会, 日本認知科学会, ACM, IEEE 等会員。博士(工学)。