

コミュニケーションロボットのための センサデータ可視化ソフトウェア RSV の設計と実装

中村 学† 川島 英之‡ 佐竹 聡† 今井 倫太‡§

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻

‡ 慶應義塾大学 理工学部 情報工学科

§ 科学技術振興機構 さきがけプログラム

E-mail: † manabu,kawasima,satake,michita@ayu.ics.keio.ac.jp

コミュニケーションロボットの行動プログラムにおける問題は、インタラクションをおこなうユーザの挙動をプログラム設計者が事前に予測することが困難なことにある。この問題を解決するために本論文はセンサデータ可視化ソフトウェア RSV を提案する。RSV は一連の行動プログラムの流れを俯瞰的に可視化するだけでなく、その微細構造をも抽出する。また、RSV はプログラム設計に際して最も頻繁に解析されるビデオとロボットのセンサデータとの連携を実現する。さらに RSV は従来研究では不可能であった膨大な量の標本を有限空間内に表示可能にする。

RSV: Robot Sensor Viewer for Communication Robot

Manabu NAKAMURA† Hideyuki KAWASHIMA‡ Satoru
SATAKE‡ Michita IMAI†§

†Scholl of Science for OPEN and Environmental Systems,
Faculty of Science and Technology, Keio University.

‡Department of Information and Computer Science, Faculty of Science and Technology,
Keio University.

§PRESTRO, JST

E-mail: dag manabu,kawasima,satake,michita@ayu.ics.keio.ac.jp

One of the most important problem to design behavior programs for communication robots is the difficulty to estimate users' behavior who interactt with robots. To tackle this problem this paper presents a novel sensor data visualizer RSV. At first, The RSV visualize both the abstraction of a sequence of behaviour programs and the microscopic structures of them. Then, the RSV coordinates sensor data of robot with video data that is analyzed most frequently for the design. Furthermore, RSV enables to display enormous samples in a limited space at the same time, that was not realized previous works.

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、ロボット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環境を認識する。現在、行動プログラムは設計者が人間の行動を予測して作成している。しかし、人間とロボットのコミュニケーションでは、人間はしばしば設計者が見落した行動を取る。行動プログラムは設計者が見落した行動に対応していないため、人間とロボットの間でコミュニケーションが破綻する。そのため、設計者が見落した人間の行動を発見するために、ロボットを人間とコミュニケーションさせて得たデータから人間の行動を分析する必要がある [1][2][3]。

人間とロボットのコミュニケーションの分析方法として一般的なものにビデオデータの閲覧が挙げられる。また、ビデオデータとセンサデータを用いる分析を補助する目的で、R-support[4]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した。

しかし、従来手法で設計者が見落した人間の行動を発見しようとすると労力がかかる。設計者が見落した人間の行動は頻繁に発見されるものではない。それゆえ、ビデオデータを見続けて人間の行動を探すと非効率的である。R-supportはビデオデータとセンサデータを同時に見ながら解析することを主な目的としているため、人間の行動分析にかかる時間はビデオデータを見続ける手法と変わらない。

そこで、この時間の問題を解決するために、次の手順を用いてデータを解析し、設計者が見落した人間の行動を発見する。センサデータの中からロボットが特定の行動をした部分を多数集める。集めたセンサデータを可視化する。可視化されたセンサデータから特徴的な部分を見つける。特徴的な部分について実際に人間がとった行動をビデオデータから確認する。この手順を用いると、あらかじめビデオデータを見ればよい部分を抽出できるため、R-supportよりも短時間で分析ができる。本研究ではこれら一連の手順を支援するセンサデータ可視化ソフトウェア RSV を提案する。

RSVはロボットが特定の行動をしたときのセンサデータを集めるデータ選択機能として、行動モジュール選択機能を提供する。RSVは集め

たセンサデータを可視化する機能を提供し、解析者による特徴的な部分の発見を補助をする。そして、RSVは特徴的な部分について、ビデオデータを再生させる機能を提供する。RSVはデータ選択機能を提供することで、ロボットの特定の行動データを短時間で抜き出せる。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、設計者が見落した行動の発見を支援する。発見した行動を設計者が行動プログラムに反映させることで、より良い人間とロボットのコミュニケーションを実現できる。

論文の構成は以下の通りである。第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。第3章で RSV の提案を述べる。第4章で RSV の設計を説明する。第5章で RSV の実行例を、第6章で RSV の考察を、第7章で結論を述べる。

2 背景とソフトウェアの要件

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。本研究ではコミュニケーションロボット Robovie[1]を対象として RSV を設計する。

2.1.1 センサデータ

Robovieは周囲の情報を認識するために、多数のセンサを使用している。Robovieは24個の距離センサ、16個のタッチセンサ、10個のバンパセンサを持つ。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は多数のセンサを使用しているため、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周期で取得している。Robovieを1時間稼働させたとなると、そのセンサデータの数には600万個にもなる。

2.1.2 行動モジュール

Robovieの行動プログラムは多数の行動モジュールから構成される。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

3.2 グラフ表示機能

グラフ表示機能は、取得した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。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。

3.2.1 距離センサグラフ

距離センサグラフは標本における距離センサデータを一括してグラフ化したものである(図2)。

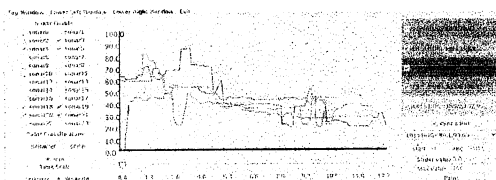


図 2: 距離センサグラフ

距離センサグラフは C_2 を満たすために距離センサグラフ表示を提供する。距離センサグラフ表示とは、すべての標本について、距離センサグラフを一括して表示する機能である。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。

これ以外にも、距離センサグラフは解析を支援するためにビューの指定、任意のグラフの非表示、タイムライン、縦軸の変更、時間軸の変更を提供する。ビューの指定は、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、標本に関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種類を提供する。任意のグラフの非表示は、任意の標本と任意の距離センサを無視した可視化するように設定する機能である。タイムラインは、距離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。縦軸の変更は、縦軸の最大値を設定する機能である。時間軸の変更は、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。

3.2.2 タッチセンサグラフ

タッチセンサグラフは各標本におけるタッチセンサデータをグラフ化したものである(図3)。

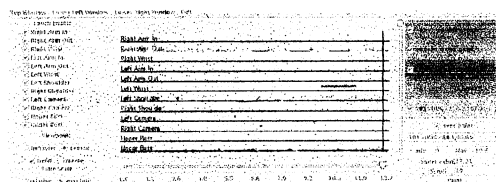


図 3: タッチセンサグラフ

タッチセンサグラフは C_2 を満たすために、タッチセンサグラフ表示、グラフのスクロールを提供する。タッチセンサグラフ表示は、いくつかの標本について、タッチセンサグラフを表示する機能である。タッチセンサデータは2値データであるため、距離センサグラフと同様に折れ線グラフを用いて重ねて表示すると、グラフが複雑化する。そのため、グラフを並列して表示しなければならない。そこで、タッチセンサグラフは帯グラフを並列してタッチセンサデータを可視化する。

グラフのスクロールとは、一度に表示しきれなかったタッチセンサグラフを表示する機能である。タッチセンサグラフはグラフを並列に表示するため、標本すべてを一括して可視化することができない。そこで、グラフをスクロールさせ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、タッチセンサグラフは解析を支援するために、ビューの指定、ビューのオプション、任意のグラフの非表示、タイムライン、時間軸の変更を提供する。

ビューの指定では、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づらくなる(図4(a))。そのため、ビューのオプションは標本やタッチセンサの種類に関係なく、反応があったことのみを検出する見やすいグラフを提供する(図4(b))。

任意のグラフの非表示、タイムライン、そして時間軸の変更は、距離センサグラフのそれと同様である。

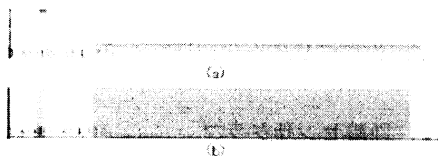


図 4: ビューのオプション

3.3 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 はビデオを搭載しておらず、画像データはカメラによる静止画像のみしか取得できない。そこで、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。従って、RSV は C_3 を満たす。



図 5: カメラ画像表示

4 設計

4.1 全体の構成

RSV の全体の構成を図 6 に示す。RSV は行動モジュール選択機能、グラフ表示機能、カメラ画像表示機能で構成され、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。

例えば、解析者がデータから握手モジュールを抜き出し(A)、握手モジュールのタッチセンサデータを可視化して(B)、その後カメラデータを見る(C)とする。この場合、システムは次

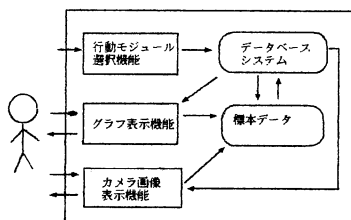


図 6: 全体の構成

のように動く。

A-(1) 解析者が行動モジュール選択機能で行動モジュールを選択する。

A-(2) 行動モジュール選択機能は、データベースシステムから選択された行動モジュールに関するデータを抜き出し、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一つ)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、標準データを用いて、データベースシステムにクエリを投げ、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、標準データを用いて、データベースシステムにクエリを投げ、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。

4.2 設計

図 6 に示すように、RSV は行動モジュール選択機能、グラフ表示機能、カメラ画像表示機能から構成される。

4.2.1 行動モジュール選択機能

行動モジュール選択機能は、データ ID 指定、行動モジュール選択、行動モジュール選択の解除、同名モジュール一括選択、行動モジュール遷移グラフ、そして行動モジュール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。ただし、図 7 では紙面の都合でいくつかの機能名を略して記述している。

```

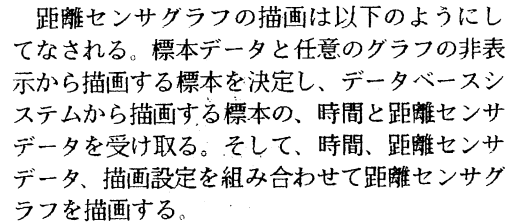
graph TD
    Start([起動時]) -- A --> DB[データベースシステム]
    DB -- A --> Type1([実動データの種別])
    DB -- B --> Type2([行動モジュールの種類])
    Type1 -- B --> Spec1[実動データ指定]
    Type2 -- B --> SelTime[行動モジュール名  
選択時刻]
    Spec1 -- B --> SelMod[行動モジュール選択]
    SelTime -- B --> SelMod
    SelMod --> Release[選択解除]
    Release --> StdData[標準データ]
    StdData --> BatchSel[一括選択]
    BatchSel --> GraphDisp[遷移グラフ表示]
    GraphDisp --> RangeSpec[描画範囲指定]
    RangeSpec --> StartTime[描画開始時刻]
    RangeSpec --> EndTime[描画終了時刻]
    StartTime --> BatchSel
    EndTime --> BatchSel
  
```

RSVは起動時にデータベースシステムに存在するデータIDを調べ、メモリに保存しておく。そして、行動モジュール選択機能を表示する際にデータIDを解析者に示す。

行動モジュール選択で解析者が時系列に提示された行動モジュールを選択すると、データ ID、行動モジュール名、行動モジュールの開始時刻、行動モジュールの終了時刻がメモリに保存される。以後、これらのデータをまとめて標本データと呼ぶ。行動モジュールデータがメモリに保存されることで選択した行動モジュールでグラフが描画されるようになる。

行動モジュール選択の解除では、解析者が選択した標本データを破棄する。行動モジュール遷移グラフでは、遷移グラフの描画開始時刻と描画終了時刻、行動モジュール名とその遷移時刻から、選択したデータ ID での行動モジュール

距離センサグラフは、距離センサグラフ表示、ビューの指定、任意のグラフの非表示、タイムライン、縦軸の変更、時間軸の変更から構成される。ここで、ビューの指定、縦軸の変更、時間軸の変更による設定を描画設定と呼ぶ。各機能とデータの流れを図8に示す。ただし、図8では紙面の都合でいくつかの機能名を略して記述している。



タッチセンサグラフはタッチセンサグラフ表示、グラフのスクロール、ビューの指定、任意のグラフの非表示、タイムライン、時間軸の変更、ビューのオプションから構成される。ここで、ビューの指定、任意のグラフの非表示、時間軸の変更、ビューのオプションによる設定を、描画設定と呼ぶことにする。各機能とデータの流れを図9に示す。ただし、図9では紙面の都合でいくつかの機能名を略して記述している。

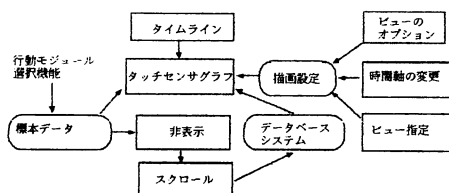


図 9: タッチセンサグラフの構成

タッチセンサグラフの描画は以下のようにしてなされる。標準データ、グラフのスクロール、任意のグラフの非表示から描画する標本を決定し、データベースシステムから描画する標本の、時間とタッチセンサデータを受け取る。そして、標準データ、時間、タッチセンサデータ、描画設定を組み合わせることでタッチセンサグラフを描画する。

グラフのスクロールは、選択した標本の中で、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。

ビューの指定、任意グラフの非表示、タイムライン、時間軸の変更は距離センサグラフと同じである。ビューのオプションは、解析者が指定したオプションに設定する。

4.2.4 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 10 に示す。

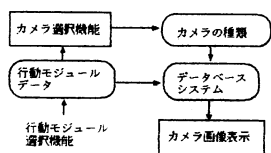


図 10: カメラ画像表示の構成

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

カメラ画像表示は、解析者が指定した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。動画として再生する際は、動画として再生するためのスレッドを作成し、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。動画を停止する際は、作成したスレッドを停止して、アニメーションを止める。

4.2.5 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 3000 行程度を要した。

5 実行例

行動プログラム設計者が見落した人間の行動を RSV が検出できることを確かめるために、5 人の被験者に 5 分間 Robovie の相手をさせた。この 5 分間の間、Robovie の行動モジュールである KISS を何回か実行させた。この KISS モジュールで人間がどのような行動をしたかを解析する。

KISS モジュールとは、まずロボットが「キスして」と発話する。次に、画像処理で人間の顔検出を試みる。そして、人間の顔が検出されるとロボットが「好き」や「大好き」と発話する行動モジュールである。

KISS モジュールを解析するために、行動モジュール選択機能でデータの中から KISS モジュールを抜き出す。抜き出せた KISS モジュールの数は 27 個である。

距離センサデータから人間の行動を発見できるかもしれないので距離センサグラフを表示する。ここで、人間はロボットの前方にいると仮定し、ロボットの前方の距離センサ以外を非表示にする。さらに、最も人間が近付いたと検知したデータ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。すると、KISS モジュールを開始してから距離センサデータが小さくなったことがわかる。ここから、人間が近付いたことが推測できる。



図 11: 距離センサグラフ

タッチセンサデータから人間の行動を発見できるかもしれないのでタッチセンサグラフを表示する。各標本ごとに可視化するビューで見たところ、いくつかの標本でタッチセンサの反応があった。そこで、どのタッチセンサが押されたかを見るために、各タッチセンサごとに可視

化するビューで見る(図12)。ここから、右肩、左肩、右目、左目のタッチセンサが押されることがわかる。

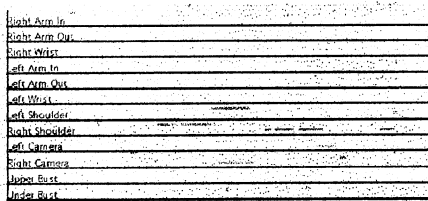


図 12: タッチセンサグラフ

以上より、KISS モジュールを行ったときに人間が近付くこと、特定のタッチセンサが押されることを検出できた。これは KISS モジュールの設計者が予測しなかった行動である。

6 考察

RSV は C_1 を解決するために行動モジュール選択機能を提供した。行動モジュール選択機能ではデータ ID の選択と、時系列に並んだ行動モジュール名の選択が同名モジュールの一括選択の 2 ステップで膨大なデータから必要なデータのみを抜き出せる。したがって、RSV は C_1 を満たす。

RSV は C_2 を解決するために距離センサグラフ、タッチセンサグラフを提供した。これらの機能で RSV はセンサデータを自動的に可視化する。そして、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、多数の標本を可視化する。よって、RSV は C_2 を満たす。

RSV は C_3 を解決するためにカメラ画像表示機能を提供した。カメラ画像表示機能はカメラデータを動画として再生する機能を提供し、動画の再生に同期して距離センサグラフとタッチセンサグラフのタイムラインを動かす。この機能により、センサデータとカメラデータの対応付けを可能にするため、RSV は C_3 を満たす。

すなわち、設計者が見落していた人間の行動を発見する支援を RSV は可能にする。

7 結論

本稿ではコミュニケーションロボットのためのセンサデータ可視化ソフトウェア RSV を設計及び実装した。RSV は人間の行動分析を支援

するための 3 つの条件を次のように満たした。

1. 行動モジュールを時系列に提示し選択させることで、ロボットが特定の行動をしたときのセンサデータの抜き出しを実現した。2. 標本における距離センサグラフ、タッチセンサグラフを作成する機能を提供し、センサデータの可視化を実現した。3. カメラ画像表示機能により、可視化されたセンサデータの中で特徴的な部分について、実際に人間がどのような行動を行ったのかを確認できるようにした。RSV を用いることで、設計者の見落していた人間の行動を発見できる。そして、発見した行動を行動プログラムに反映させることで、ロボットと人間のコミュニケーションをより良くできると結論する。

参考文献

- [1] Takayuki Kanda, Hiroshi Ishiguro, Tet-suo Ono, Michita Imai and Ryohei Nakatsu: "Development and Evaluation of an Interactive Humanoid Robot "Robovie", " IEEE International Conference on Robotics and Automation (ICRA 2002), pp.1848-1855, 2002.
- [2] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, and Hiroshi Ishiguro: "A practical experiment with interactive humanoid robots in a human society," IEEE International Conference on Humanoid Robots, Oct. 2003.
- [3] 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, pp2699-2709, 2003.
- [4] 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.