

# 人の運動解析のための3次元表示に関する研究

田房友典\*・安藤美紀\*\*

## Three-Dimensional Display System for Analyzing Human Motion

Tomonori Tabusa\* and Miki Ando\*\*

### Abstract

We proposed three-Dimensional recovery based on extended measurement matrix. This matrix decomposes into camera position and 3-D coordinate by factorization. This way does not need camera calibration, so it is very useful for analyzing human motion and some medical applications. This paper proposes novel 3-D display system of position data obtained by 3-D recovery employing factorization. We use wire frame model and resold cube model to connect of position data.

### 1. はじめに

コンピュータ技術の進展により、ビジョンセンサや画像処理システムが非常に安価に利用可能になってきた。それにともない、物体の形状や動きを3次元復元する技術も進歩し、スポーツ分野では運動解析、医学や福祉の分野ではリハビリにおける訓練効果の確認等、様々な分野での応用が期待されるようになってきている。

一般的環境における3次元復元は、測定対象につけられた特徴点の画像上の座標値を、ステレオ法やタンラ<sup>[1]</sup>の拡大計測行列に基づく復元法を用いて3次元座標に変換する。3次元形状に変換された特徴点の座標値の羅列だけでは、正確に復元されているのか判断することや運動を解析することは困難である。通常、3次元データの再現には、テクスチャを貼り付け、よりリアルに再現する手法が用いられている。テクスチャの貼り付け手法に、市販のソフトウェアを用いる。市販のソフトウェアを用いて人のリアルな再現を行う場合、関節の個数以上の特徴点が必要となる。本研究では、光学式復元法であるステレオ法や拡大計測行列に基づく復元法で、3次元復元する手法において、復元された任意の数の特徴点座標をモデル化し、ディスプレイ上に再現するソフトウェアの開発を行う。

光学的復元法の中でもカメラキャリブレーションを必要とせず、多視点画像だけを用いて変形体を3次元座標に変換する手法は、一般的環境下では汎用性が高い<sup>[2]</sup>。このため、拡大計測行列に基づく復元法によって得られた復元結果を再現するソフトウェアの開発にターゲットを絞る。

拡大計測行列に基づく復元法によって3次元復元された特徴点座標は、1フレーム毎の特徴点が時間軸方向に列挙されたデータ形式である。このデータをワイヤーフレームモデルとソリッドモデルを用いて3Dグラフィックス描画を行う。開発ソフトウェアの特徴は、3次元復元された特徴点数と形状に制約を受けず、ユーザが任意の形状にモデル化可能な点である。ソフトウェアの実現には、OpenGLライブラリを用いて、復元データである3次元特徴点座標ファイルに加え、ユーザが作成する接続情報を記述した接続情報ファイルを入力することでモデル化の形状を決定する。

実験として、人の運動に対して拡大計測行列に基づく変形体の3次元復元法<sup>[2]</sup>を適用し、復元された特徴点の3次元座標を開発ソフトウェアによってモデル化し、評価、考察を行う。

### 2. 拡大計測行列に基づく運動の3次元復元法

$F (F \geq 3)$  台のビデオカメラを設置し、計測対象となる人を撮影する。そのとき  $f (f = 1, 2, \dots, F)$  番目のビデオカメラの映像  $I_f(t) (t = 1, 2, \dots, T)$  によって、特徴点  $s_p(t) (p = 1, 2, \dots, p_i)$  が観測される。この特徴点の映像上での位置を  $(x_{fp}(t), y_{fp}(t))$  とする。時刻  $t$  における  $I_f(t)$  上の  $xy$  座標を順に、対応を取りながら計測行列  $W(t)$  に記入する。

\* 情報工学科

\*\* JR 西日本

$$W(t) = \begin{pmatrix} x_{11}(t) & x_{12}(t) & \dots & x_{1p}(t) \\ x_{21}(t) & x_{22}(t) & \dots & x_{2p}(t) \\ \vdots & \vdots & \vdots & \vdots \\ x_{F1}(t) & x_{F2}(t) & \dots & x_{Fp}(t) \\ y_{11}(t) & y_{12}(t) & \dots & y_{1p}(t) \\ y_{21}(t) & y_{22}(t) & \dots & y_{2p}(t) \\ \vdots & \vdots & \vdots & \vdots \\ y_{F1}(t) & y_{F2}(t) & \dots & y_{Fp}(t) \end{pmatrix} \quad (1)$$

計測行列  $W(t) (t = 1, 2, \dots, T)$  をまとめれば、式(1)に示す拡大計測行列  $W$  が得られる。

$$W = (W(1) | W(2) | \dots | W(T)) \quad (2)$$

さらに  $W$  を、次式を用いて行列  $\tilde{W}$  に変換する。

$$\tilde{W} = W - \frac{1}{Q} W \cdot E \quad (3)$$

ここで、 $Q$  は、行列  $W$  に記入された全特徴点の数、 $E$  は成分がすべて1の  $Q \times Q$  行列である。行列  $\tilde{W}$  は、階数がたかだか3であり、各ビデオカメラの方向を与える姿勢行列  $M$  と、全特徴点の3次元座標を与える形状行列  $S$  とに分解することができる。

$$\tilde{W} = M \cdot S \quad (4)$$

この分解を実際に行うには、 $\tilde{W}$  を特異値分解し、カメラレンズ座標系の正規直交性を利用して唯一解を求める。

### 3. 3次元グラフィックス表示

特異値分解によって分解された3次元座標を与える形状行列  $S$  は、物体に付けられた特徴点の3次元座標をフレーム毎に列挙したデータ配列である。(3.1章参照)。データ配列だけでは、フレームの区切りや物体形状を把握することは困難である。

形状行列  $S$  は、特徴点が記述されている順序が各フレームで同じである。形状行列  $S$  を示す3次元特徴点座標ファイルと、フレーム毎の特徴点数や特徴点間の接続情報を記述した接続情報ファイルを用いて、任意の対象物に対して3次元グラフィックス表示を行う。

#### 3.1 3次元特徴点座標ファイル

3次元復元された特徴点座標を図3.1に示す書式で記

述する。対象が変形体のように時間軸方向にもデータが存在する場合、全ての特徴点において対応がとれていなければならない。ここで言う対応とは、各フレームにおいて特徴点の記述された順番が同じであることである。また、ファイルには、全フレームにおける特徴点が欠如することなく記述されなければならない。1フレームの特徴点数が  $P$ 、全フレーム数が  $F$  とすると、3次元特徴点座標ファイル上には、 $P \cdot F$  個の3次元座標が記述されている。特徴点の  $x, y, z$  座標の表記は整数値、区切りはスペースもしくはタブである。

#### 3.2 接続情報ファイル

対象物の特徴点を接続する情報を接続情報ファイルに記述する。特徴点は、表3.1に示すように、直線、三角形、四角形と円で接続することが可能である。線分は、始点と終点の特徴点番号を指定し、順不同である。円は、中心となる特徴点番号、半径と色を指定する。三角形、四角形は、それぞれを構成する特徴点番号および色を指定する。円、三角形、四角形で指定する色は、塗りつぶし色であり、引数と色の関係は、表3.2に示す。

特徴点が  $P$  個で対象物が構成されていると、接続情報ファイルに記入可能な特徴点番号は、1 ~  $P$  である。接続情報ファイルは、図3.2に示すように、1行目に対象物に付けられた特徴点数 ( $P$ ) を記入する。2行目以降は、各行に対象物を構成する形状を1個ずつ記入する。各特徴点番号や色の区切りにはスペースを用い、各行に

| $x$ 座標 | $y$ 座標 | $z$ 座標 |          |
|--------|--------|--------|----------|
| 80     | 41     | -36    | 1 フレーム   |
| 57     | 41     | -41    |          |
|        | ⋮      |        |          |
| 80     | 42     | -33    | 2 フレーム   |
| 55     | 40     | -41    |          |
|        | ⋮      |        |          |
| 81     | 41     | -30    | 3 フレーム   |
| 57     | 39     | -41    |          |
|        | ⋮      |        |          |
|        | ⋮      |        |          |
|        | ⋮      |        | $F$ フレーム |

図3.1 3次元特徴点座標ファイル

表3.1 描画可能な形状と書式

| 形 状   | 書 式                               |
|-------|-----------------------------------|
| 線 分   | 特徴点番号 1 特徴点番号 2                   |
| 円     | 特徴点番号 半径 色                        |
| 三 角 形 | 特徴点番号 1 特徴点番号 2 特徴点番号 3 色         |
| 四 角 形 | 特徴点番号 1 特徴点番号 2 特徴点番号 3 特徴点番号 4 色 |

表 3.2 引数と塗りつぶし色

| 引数 | 色    | 引数 | 色  |
|----|------|----|----|
| 0  | 黒    | 8  | 灰色 |
| 1  | 暗い青  | 9  | 青  |
| 2  | 茶    | 10 | 赤  |
| 3  | 暗い赤茶 | 11 | 赤紫 |
| 4  | 暗い緑  | 12 | 緑  |
| 5  | 暗い青緑 | 13 | 青緑 |
| 6  | 暗い黄色 | 14 | 黄  |
| 7  | 暗い灰色 | 15 | 白  |

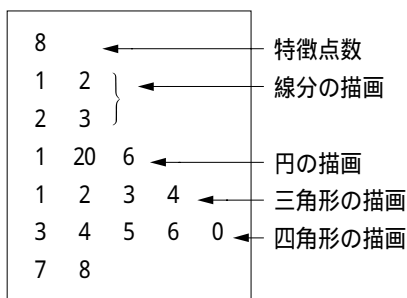


図 3.2 接続情報ファイル

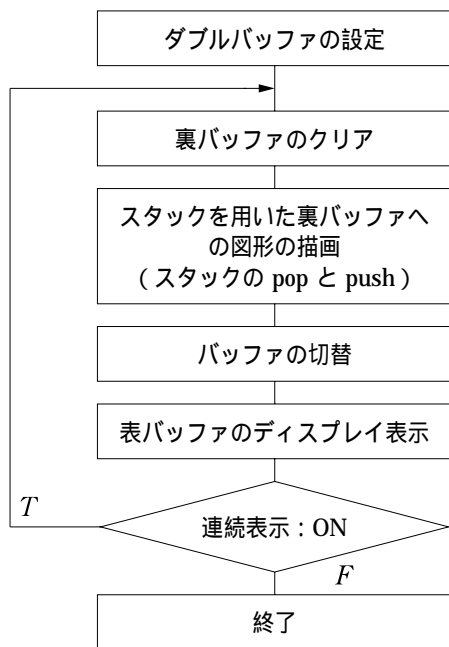


図 3.3 OpenGL を用いた 3次元グラフィックスの描画

記入する形状の順位は任意である。

### 3.3 OpenGL を用いた 3次元グラフィックスの実現

3次元グラフィックスの実現には、OpenGL を用いる。変形体の形状復元では、アニメーションのように頻りに画面を書き換える必要がある。このため、ダブルバッフ

ァ機能を用いてスムーズな描画を実現する。ダブルバッファは、2画面分のフレームバッファを用意しておき、一方の画面を表示している間、裏に回っているフレームバッファに次の絵を描画し、描画が完了した時点で2つのバッファを切り替える方法である。

また、複数の図形の描画や処理の高速演算を実現するために、行列スタックを用いる。OpenGL にはローカル座標の状態(モデルビュー行列)を記憶する行列スタックが容易されており、スタックに現在のモデルビュー行列を記憶させたり、取り出しを行ったりすることにより、さまざまな視点から複数の図形を描画することができる。スタックは、最低でも32段のモデルビュー行列スタックが利用することができる<sup>[3]</sup>。

3次元グラフィックスを実現するプログラムの流れを図 3.3 に示す。ただし、破線で囲まれた領域は、ユーザからの繰り返し判定が、「真」の間、描画を行う。

## 4. システム

### 4.1 インストール

開発ソフトウェアは、3.5インチフロッピディスクに格納され(1.29Mbyte)、「3 dv.exe」という自己解凍形式ファイルをクリックするだけで各クライアント PC にインストールすることが可能である。

「3 dv.exe」を解凍すると、表 4.1 に示すようなファイルが展開される。ここで、「3 dv.exe」は、プログラム本体、「bsk.txt」はサンプルデータの3次元座標ファイル、「info\_bsk.txt」は、サンプルデータの接続情報ファイルであり、その他のファイルは、開発ソフトウェアの実行に必要な環境ファイルである。また、本開発ソフトは、Windows95/98/98SE/ME/2000で動作を確認済みである。

表 4.1 開発ソフトの構成ファイル

| ファイル名        | 役割                   |
|--------------|----------------------|
| 3 dv.exe     | プログラム本体              |
| bsk.txt      | サンプルファイル3次元座標データファイル |
| bsk info.txt | サンプルファイル接続情報ファイル     |
| borlndmm.dll | 環境ファイル               |
| cp3245mt.dll |                      |
| glut32.dll   |                      |
| bcbmp40.bpl  |                      |
| vcl40.bpl    |                      |
| vclx40.bpl   |                      |

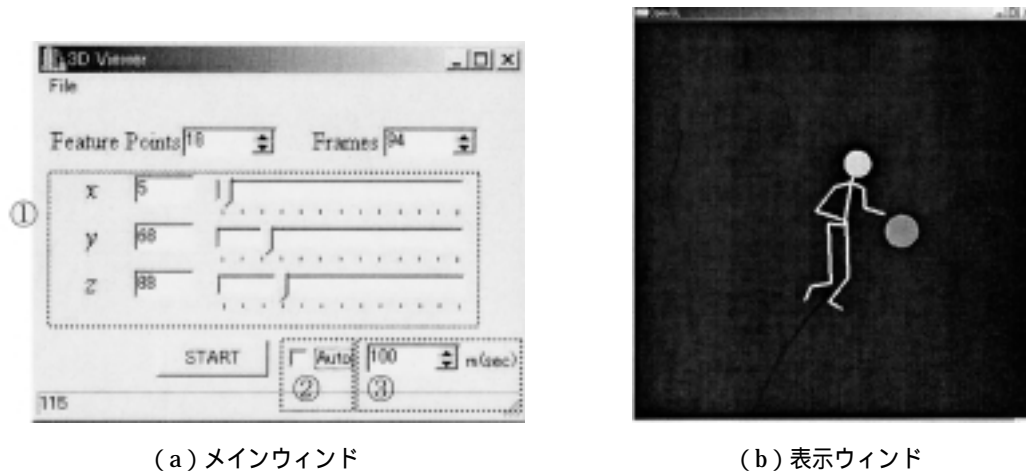


図4.1 構成ウィンド

## 4.2 構成

開発ソフトウェアは、図4.1に示すようにメインウィンドウと表示ウィンドウによって構成される。表示ウィンドウは、表示ウィンドウの中央を原点(0, 0)とし、各3次元座標軸ともに-255~255までの座標値まで表示可能である。同図(a)において①を表示視点角度調整領域と呼び、トラックバーとエディットボックスに数値を設定可能である。②は接続表示チェックボックス領域、③は表示間隔設定領域と呼ぶ。

## 4.3 機能

### 4.3.1 表示視点角度調整

$x, y, z$  軸を0~359°まで、回転可能である。回転角度の変更は、トラックバーとエディットボックスに数値を入力することで行う。表示ウィンドウの表示前に視点角度を調整すれば、表示後の物体の視点角度に反映される。また、連続表示チェックボックスのオン/オフにかかわらず、調整可能である。

### 4.3.2 連続表示

連続表示のオン/オフ設定を行う。チェックボックスにチェックが付いている状態が連続表示オンである。表示対象が0~P枚のフレームで構成されているとき、P枚目のフレーム表示後、0枚目のフレームへと表示を繰り返す。

### 4.3.3 表示間隔調整

連続表示がオンであるとき、表示間隔を1~9,999,999msecまで設定する。設定は、連続表示チェックボックスのオン/オフにかかわらず可能である。

## 4.4 使用方法

開発ソフトウェアの使用手順を図4.2に示す。初期パラメータの設定では、各軸( $x, y, z$ )の表示視点角度、連続表示チェックボックス、表示間隔の設定を行う。各

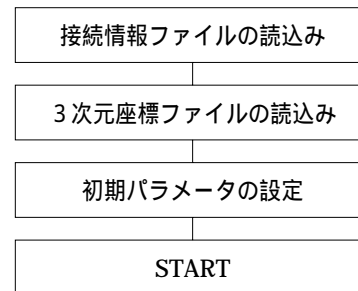


図4.2 操作手順

表4.2 各種パラメータの設定値

| パラメータ               | 値         |
|---------------------|-----------|
| 表示視点角度( $x, y, z$ ) | (0, 0, 0) |
| 連続表示チェックボックス        | OFF       |
| 表示間隔                | 100ms     |

初期パラメータの設定値を表4.2に示す。

## 5. 実験結果

人の運動を3台のビデオカメラで測定し、表5.1に示す3次元形状Sを開発ソフトウェアによって再現した。各対象をモデル化するために用いた接続情報ファイルを図5.1、測定1~測定3の復元結果を開発ソフトウェアによって再現した結果を図5.2~図5.4に示す。同図

表5.1 各種パラメータの設定値

| 測定No | 運動概要         | 特徴点数 |
|------|--------------|------|
| 測定1  | 歩行する人        | 15   |
| 測定2  | バスケットボールをする人 | 18   |
| 測定3  | ローラーブレードをする人 | 19   |

|  |  |  |
|--|--|--|
| <pre> 15 1 18 9 1 2 2 3 10 9 6 3 4 4 5 2 6 13 9 6 6 7 7 8 10 11 11 12 13 14 14 15                 </pre> | <pre> 18 1 2 2 3 3 4 4 5 2 6 6 7 7 8 2 9 10 9 10 11 11 12 12 13 9 14 14 15 15 16 16 17 18 22 2 1 18 8                 </pre> | <pre> 19 1 2 2 3 3 4 4 5 2 6 6 7 7 8 2 9 10 9 10 11 11 12 12 13 14 3 9 15 15 16 16 17 17 18 19 3 1 18 8                 </pre> |
| (a) 測定1  | (b) 測定2  | (c) 測定3  |

図5.1 接続情報ファイル



図5.2 グラフィック表示

は、表示移転角度  $(x, y, z)$  を  $(0, 77, 77)$  に設定した結果である。

## 6. おわりに

複数台のビデオカメラで撮影した被写体(変形体)の特徴点座標は、タンらの方法によってカメラパラメータ  $M$  と3次元形状  $S$  に分解することができる。被写体の3次元形状を表す  $S$  をディスプレイ上にモデリングするソフトウェアの開発を行った。

本ソフトウェアは、復元された運動を一時停止/スロ

ー再生/回転および、必要なフレームを瞬時に表示することが可能であるため、運動の解析・分析を行うときに補助的に利用することができる。また、被写体の3次元形状を表す3次元特徴点座標ファイルに加え、被写体の接続情報を表す接続情報ファイルを入力することにより、いかなる形状の被写体であっても柔軟にワイヤーフレームによって接続し、形状をモデル化することができる。ワイヤーフレームだけを用いた形状のモデル化だけでなく、球、三角形、四角形の図形を用いて形状を構成することも可能である。着目したい関節や部位においては、色を変えることによって解析の補助や、ユーザに対して

運動の特徴を強調することができる。

本ソフトウェアは、3次元形状  $S$  の結果を表示し、奥行き等の復元状態を確認するためには、操作も容易であり十分な役割を果たす。今後の課題として、解析や分析などに利用する場合に、関節や特定部位の動きを解析する軌跡表示機能、移動距離を測定するためのグリッド表示機能、復元対象の座標値のスケールが表示ウィンドウに合わない場合に、拡大/縮小機能が必要である。

### 参考文献

- [ 1 ] タン, 石川, 加藤: 因数分解を利用したモーションキャプチャ法, 計測自動制御学会論文集, 36 - 11, 980 / 984 (2000).
- [ 2 ] T. Tabusa, J. K. Tan, S. Ishikawa: Recovering Human Motions by Mobile Cameras and Factorization, The 5th Asian Conference on Computer Vision, 551 - 555, 2002.
- [ 3 ] 酒井: OpenGL 3D プログラミング - 3DCG の基礎から制御系アニメーションへの応用まで, CQ 出版 (2000).
- [ 4 ] クレイトン: Win32OpenGL プログラミング Windows NT / 95 3次元グラフィックスプログラミング入門, ピアソンエデュケーション (1999).
- [ 5 ] 田中, 手塚: C++Builder 4 コンポーネント活用ガイド&実践プログラミング Vol.1 プログラミング・テクニック, カットシステム, (1999).
- [ 6 ] 桐山: C++Builder による Windows プログラミング, 森北出版 (1999).