グラフィクスとCAD 79－3
（1996. 2. 23）

二次元ドローデータによる三次元多関節構造体のモデリング

新井 清志　　堀井 洋一

（株）日立製作所　中央研究所
〒185 東京都国分寺市東恋ヶ窪1-280

コンピュータグラフィックスを用いた三次元アニメーションは、映像表現の一手法として定着した。しかし、三次元アニメーションを制作する際、三次元形状作成の難しさが、特に素人にとって問題となっていた。本報告では、二次元図形のベクトルデータと文字列データとから成る二次元ドローデータから、図形の傾きや文字列の内容を手がかりとして三次元多関節構造体を自動生成する手法を提案する。本手法により、三次元形状のモデリング作業を大幅に簡素化することができた。

MODELING OF 3D ARTICULATED FIGURES BY 2D DRAWING

Kiyoshi Arai　　youichi Horry

Central Research Laboratory, Hitachi, Ltd.
1-280 Higashi-Koigakubo, Kokubunji, Tokyo 185, Japan

Three-dimensional computer animation has become the common technique of generating moving pictures. However, 3D modeling in making 3D computer animation is difficult, especially for non-professional users. In this paper, we propose the method of modeling 3D articulated figures using "2D drawing data" consisting of vectors representing 2D shapes and characters. By referring to the directions of the shapes and the description of the characters, the 3D articulated figures are automatically generated. This method simplifies 3D modeling to a great extent.

1. はじめに

三次元のコンピュータグラフィックス（ＣＧ）を用いたアニメーションは、映画、テレビ番組、あるいはゲーム等の制作における映像表現の一手法として定着した。三次元ＣＧアニメーション制作のプラットフォームは主としてワークステーションだが、パーソナルコンピュータ（ＰＣ）も処理性能の向上によって新たな道具の一つとなりつつある。一方、インターネットの世界では、三次元ＣＧデータを記述する標準言語が登場した[1]。今後、ＰＣで制作した三次元ＣＧアニメーションを、インターネット上での情報発信の素材等に利用したいというニーズが高まると予想される。このとき、アニメーション制作に必要な三次元形状モデリング作業の難しさが、特に素人にとって問題になると考えた。本報告では、この作業を簡素化する三次元形状生成手法を提案する。

2. 三次元形状作成の課題

三次元ＣＧアニメーションにおいて最も一般的に用いられる三次元形状は、階層構造を有する三次元多関節構造体であり、その関節角の変化によってアニメーションに必要な動きが得られる。三次元多関節構造体のモデリング作業は、部品となる三次元形状の作成作業と、作成した三次元形状を階層構造化する作業から成る。いずれの作業過程も、三面図、すなわち互いに直交する三方向からの二次元表示と、遠近感の付いた三次元表示とによってユーザに示される[2][3]。階層構造化は、部品となる三次元形状を１つずつ配置して組み立てる方法か、または予め最終的な位置に配置されている三次元形状に対して「骨格」となるローカル座標系を指定する方法によって実現される。

このような作業において、ユーザが所望の三次元形状を思い浮かべる場合、１枚の二次元図形が「厚み」を持ったものとして捉えることが多く[4]、複数の表示を結び付けようとすることは、かえって三次元形状の全体像の把握の妨げになりやすい。また、階層構造化作業においては、「骨格」を指定する方法を用いた場合、ユーザ所望のローカル座標系は、最終的な位置に配置された三次元形状の「傾き」と密接な関係にあるので、自動的に階層構造を生成できる可能性がある。

そこで、本研究では、１枚の二次元図形で表現された情報を基にして、三次元多関節構造体の形状と階層構造を自動生成することにより、モデリング作業の簡素化を試みた。１枚の二次元図形としては、多くの市販ソフトウエアが扱っており、図形に対する「言語的指示」[5]も記述できる二次元ドローデータを用いた。

3. 二次元ドローデータからの三次元形状生成

本報告で提案する三次元形状生成手法では、二次元図形のベクトルデータと文字列データとから成る二次元ドローデータを入力データとする。二次元図形に「厚み」を与えて立体化する際に必要な情報は、図形と対応関係を持つ文字列の記述で定義される。立体化された図形は、図形の「傾き」を考慮したローカル座標系を用いて階層構造化される。

3．1 二次元ドローデータの内容

本形状生成手法で用いる二次元ドローデータの例を図１に示す。このデータは、２本の腕、２本の足、上下２つに分れた胴体、目と口を持つ頭という１０個の図形から成るＣＧキャラクタを表わしている。その他に、足の左横には予備の楕円体が１個、足の下には特別な意味を持つ矩形が４個並んでおり、全部で１５個の独立した二次元図形が存在している。これらの各々を「部品」とよぶ。本手法では、面積を持つ図形のみを部品とみなす。足の上下端にある２本の線分のように面積を持たない図形は、形状生成処理の際に無視されるので、部品を配置するための補助線として利用できる。各々の部品の内部は、一定の色やパターンで塗られているものとする。

図１の中には、二次元図形の他に、文字が１行以上並んで独立した集合を成すものが、全部で１２個存在している。これらの各々を「文字列」とよぶ。部品と文字列の中から選ばれた複数の要素を、同一のグループに属するものと定義することができる。この定義を「グループ化」とよぶ。多関節構造体を生成する際、各々の部品は基本的には独立した階層の三次元形状になるが、複数の部品が同一のグループに属するときは、それらの部品は同一の階層の三次元形状になる。また、グループの中で最も背面にある部品以外のものを「副部品」とよぶ。図１の例では、頭と目と口がグループ化されているものとする。この場合、目と口は頭の副部品であり、頭と同一の階層の形状になる。

3．2 文字列の種類と役割

文字列は、同一のグループとして定義された部品か、または副部品でない部品で文字列の最も近くに存在するものとの間に対応関係を持つ。ただし、以下の２種類の文字列は部品との対応関係を持たない。

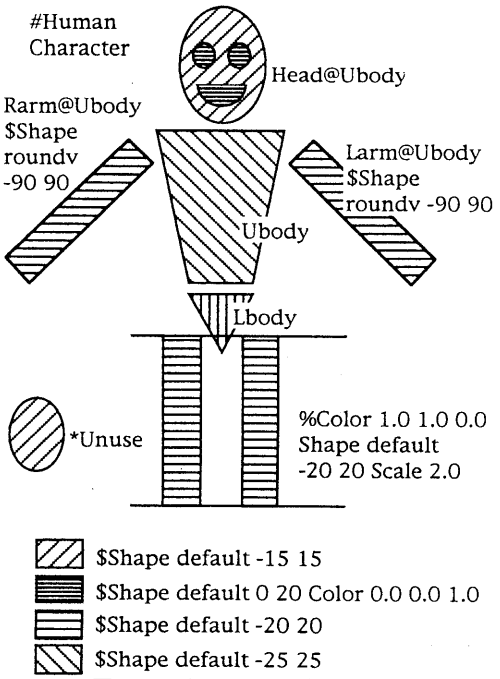


図１　二次元ドローデータの例

（１）「#」で始まる文字列は、注釈文として扱われ、無視される。図１では、文字列「#Human Character」が注釈文である。

（２）「%」で始まる文字列は、全ての部品に影響を与える初期設定文である。初期設定文の内容は、部品の色や厚みの付け方等といった「部品の属性」のデフォルトの設定、部品を立体化する際のワールド座標系の座標軸の方向やスケールの指定、変数の定義等である。図１では文字列「%Color 1.0 1.0 0.0 Shape default -20 20 Scale 2.0」が初期設定文である。

上記以外の文字列は、特定の部品との間に対応関係を持つ。対応する文字列の内容によって、部品は以下のように分類される。

（１）「*」で始まる文字列に対応する部品は、「不使用部品」として無視される。この文字列は注釈文として扱われ、無視される。図１では、文字列「*Unuse」とその左横にある予備の楕円体が対応しており、この楕円体は不使用部品となる。

（２）「$」で始まる文字列に対応する部品は、「凡例部品」となる。これらの文字列は、凡例部品と同一の色やパターンによって内部が塗られている全ての部品の属性を記述する。図１では、文字列「$Shape default -15 15」はその左横にある矩形に対応し、この矩形と同一のパターンで塗られているＣＧキャラクタの頭の属性を記述している、同様にして、文字列「$Shape default 0 20 Color 0.0 0.0 1.0」は目と口の属性を記述している。

（３）上記以外の文字列に対応し、かつ副部品ではない部品は「主部品」とよばれ、独立した階層の三次元形状になる。主部品に対応する文字列には、部品の名前、部品の親の階層となる「親部品」の名前、および部品の属性を記述する。親部品の名前は「@」の後に記述し、部品の属性は「$」の後に記述する。部品の名前以外は、無くてもよい。例えば、図１の文字列「Rarm@Ubody$Shape roundv -90 90」は、対応する部品、すなわちＣＧキャラクタの右腕の名前が「Rarm」、その親部品の名前が「Ubody」であることを意味し、部品を立体化して得られる形状が、部品の傾きの方向を中心軸とする円柱になることを「Shape roundv」というキーワードで示している。「-90 90」はこの円柱の形状に変化を与えるパラメータである。なお、対応する文字列が無く、かつ副部品でない部品も、「主部品」となる。予め文字列との対応があり、かつこれらの部品の最も近くに位置する部品が、これらの部品の親部品になる。図１では、ＣＧキャラクタの右足と左足には対応する文字列がなく、「Lbody」という名前を持つ胴体下部が親部品になる。

（４）上記のいずれにも該当しないのは、副部品だけである。副部品は立体化されて、同一のグループに属する主部品と同一の階層に置かれる。副部品に対しても、グループ化によって文字列を対応させ、その副部品の属性を記述することができる。

主部品および副部品の各々の属性は、複数の文字列によって重複して記述される可能性がある。これらの記述の参照は、以下の優先順位で行なわれる。

（１）まず、部品に対応する文字列の中に属性の記述があれば、これを参照する。

（２）次に、部品と同一の色やパターンによって内部が塗られている凡例部品があれば、この凡例部品に対応する文字列の中にある属性の記述を参照する。

（３）次に、「%」で始まる文字列の中に属性の記述があれば、これを参照する。

（４）上記のいずれの中でも記述されていない属性に関しては、暗黙の初期設定が存在し、これが参照されるものとする。

３．３　部品の検出と文字列の検出

本形状生成手法では、二次元ドローデータの中から、まず部品と文字列を検出する。一定の色やパターンによって塗られた単体の二次元図形は、部品として検出される。部品の形状は、頂点列から成る二次

元ポリゴンで表現する。単体の二次元図形が二次元ポリゴン以外の図形、例えば楕円や扇形等であった場合は、これらの図形の輪郭に適当な間隔で頂点を並べて二次元ポリゴンを作り、この二次元ポリゴンで部品を表現する。さらに、二次元ポリゴンの頂点から、部品の中心点、中心軸、および２つの端点を求める。これらは、部品のローカル座標系を決定する際に重要な手がかりとなる。図２は、図１の二次元ドローデータの中から、ＣＧキャラクタの右腕を４角形の部品として検出する処理を示している。この処理は以下の手順で行なわれる。

（１）４角形の４頂点を標本点とする一次回帰式に基づいて、部品の「傾き」を求める。

（２）部品の傾きの方向、および部品の傾きに垂直な方向の頂点の分布範囲の中心を、部品の「中心点」とする。部品の中心点を、部品の「位置」とみなす。

（３）中心点を通り、部品の傾きの方向を向く直線を、部品の「中心軸」とする。

（４）部品の４頂点から中心軸に下した垂線の足のうち両端に位置するものを、部品の２つの「端点」とする。

文字が１行以上並んで独立した集合を成すものは、文字列として検出される。文字列の存在領域の左上の端点を、その文字列の「位置」とみなす。部品と文字列、または部品と部品の位置の隔たりを、それらの間の「距離」とみなす。

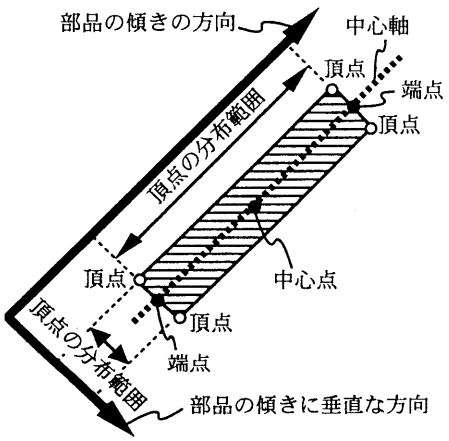


図２　部品の検出

３．４　部品の属性と階層構造の決定

検出された部品と文字列の対応付けにより、部品の属性と階層構造を以下の手順で決定する。

（１）特定の部品とグループ化されている文字列は、その部品に対応させる。

（２）対応未定の文字列の各々を、対応未定の主部品候補の中で最短距離にある部品に対応させる。

（３）対応未定の主部品候補をＭ、対応未定の副部品候補をＳとする。

（４）全てのＭに対して、新たな文字列を生成して対応させ、Ｍに対して「仮の」名前を新たに与えてこの文字列の中に記述する。（２）までの処理で対応が確定した主部品のうち、Ｍから最短距離にある主部品の名前を、Ｍの親部品の名前とし、Ｍに対応させた文字列の中に記述する。全てのＳに対しても、新たな文字列を生成して対応させる。

（５）対応確定済みの主部品をＭ、対応確定済みの副部品をＳとする。

（６）全てのＭおよびＳから、その部品と同一の色やパターンによって内部が塗られている凡例部品に対応する文字列の記述を参照できるようにする。

（７）全てのＭに対して、対応確定済みの主部品の中から、名前がＭの親部品の名前と一致する部品をさがし、Ｍから参照できるようにする。全てのＳに対して、Ｓと同一のグループに属する主部品を、Ｓから参照できるようにする。

以上の処理によって、部品の立体化に必要な属性情報と階層構造情報が確定する。

３．５　部品のローカル座標系の決定

次に、部品の三次元形状を記述するためのローカル座標系を決定する。部品の中心点および両端点のうち、親部品の中心点からの距離が最も短いものをローカル座標系の原点とし、部品の中心軸をローカル座標系の座標軸のうちの一つとする。図３は、図１のＣＧキャラクタの右腕のローカル座標系を決定する処理を示している。右腕の中心点および２つの端点の各々に対して、親部品となる胴体上部の中心点からの距離を求める。その結果、上の端点までの距離が最も短いので、これがローカル座標系の原点となる。そして、ローカル座標系の座標軸のうちの一つは右腕の中心軸の方向と一致するものになり、他の一つはこの部品を含む平面内で中心軸と直交するものになる。右腕を立体化するために必要な残りの一つの座標軸の方向は、この部品を含む平面に対して垂直な方向に定められる。なお、必要に応じて座標原点や座標軸の方向を、部品に対応する文字列で記述して指定してもよい。部品のローカル座標系は、部品を立体化する際の縦横の方向の基準として用いられる。また、主部品のローカル座標系は、部品から生成した三次元形状を記述する際にも用いられ、部品の階層構造の基準となる。

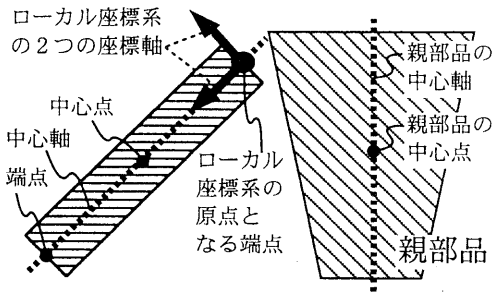


図３　部品のローカル座標系の決定

３．６　三次元多関節構造体の生成

最後に、部品の立体化処理、すなわち部品に「厚み」を与える処理を行なう。部品を変形して得られる図形を、部品を含む平面の前後に複数個重ね、重ねた図形群の輪郭が表わす三次元形状を三次元ポリゴン等で表現する。図４は、図１のＣＧキャラクタの右腕の立体化処理を示している。右腕を表わす４角形の上に２個、下に２個、計４個の図形を重ねている。これらはいずれも４角形の縦横のスケールを変化させて生成した図形である。最も上の図形と最も下の図形は、４角形の中心軸と垂直な方向のスケールがゼロであるため、縮退して線分になっている。これら合計５個の図形の輪郭は、２つの８角形を両端とする８角柱を形成する。これを三次元ポリゴン等で表現することにより、視覚的に円柱とみなされる三次元形状が得られる。このような立体化手法により、円柱、角柱、円錐、角錐、円錐台、角錐台、楕円体等の様々な三次元形状を生成できる。これらの形状は、部品を含む平面に対して対称な位置に２個生成してもよい。また、これらの形状を、あらかじめ用意された他の三次元形状と差し替えてもよい。生成する三次元形状の種類、すなわち厚みの付け方は、部品に対応する文字列で記述して指定する。

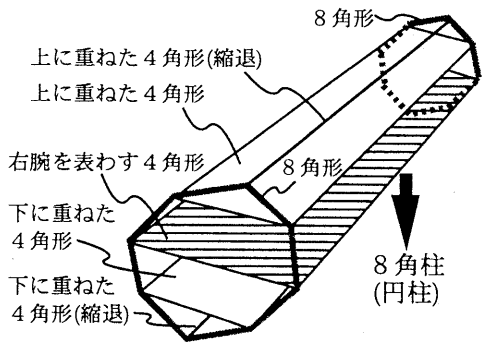


図４　部品の立体化

このようにして立体化された部品が、各々の親部品の子供として階層化され、三次元多関節構造体が生成される。図１のＣＧキャラクタは、図５に示すような階層構造を持つ三次元多関節構造体になる。「Root」は親部品が存在しない部品「Ubody」「Lbody」に対する仮想的な親、「U-1」「U-2」は両足に与えられた仮の名前である。図１のＣＧキャラクタから生成した三次元多関節構造体のフラットシェーディング画像を図６に示す。

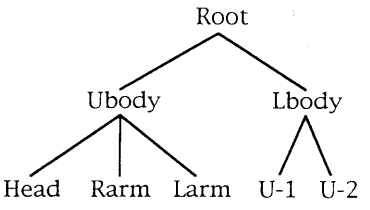


図５　ＣＧキャラクタの階層構造

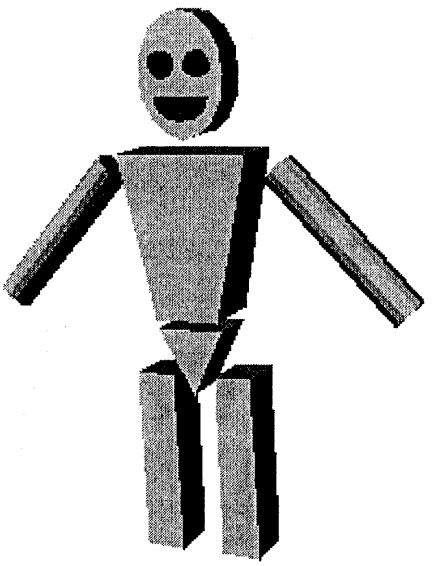
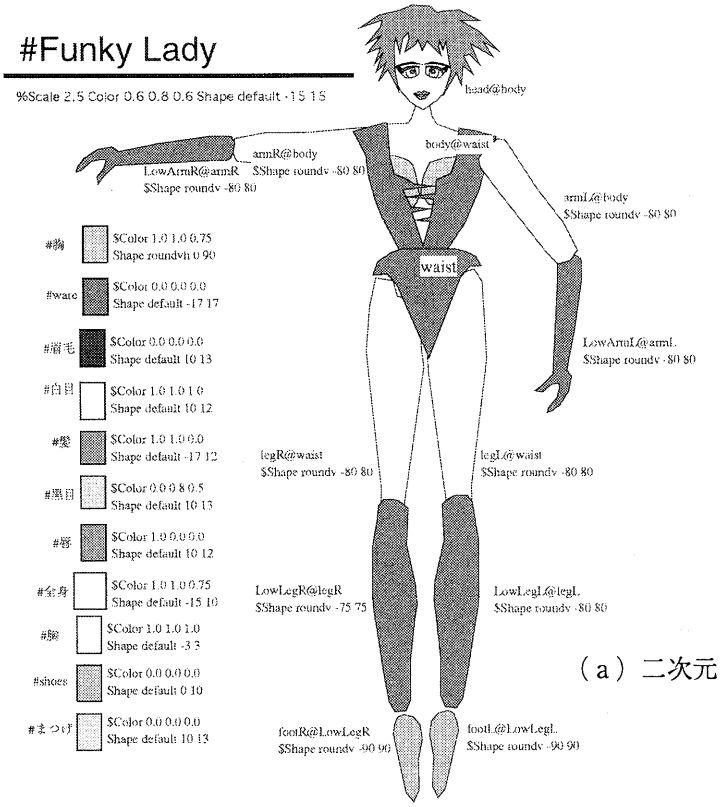


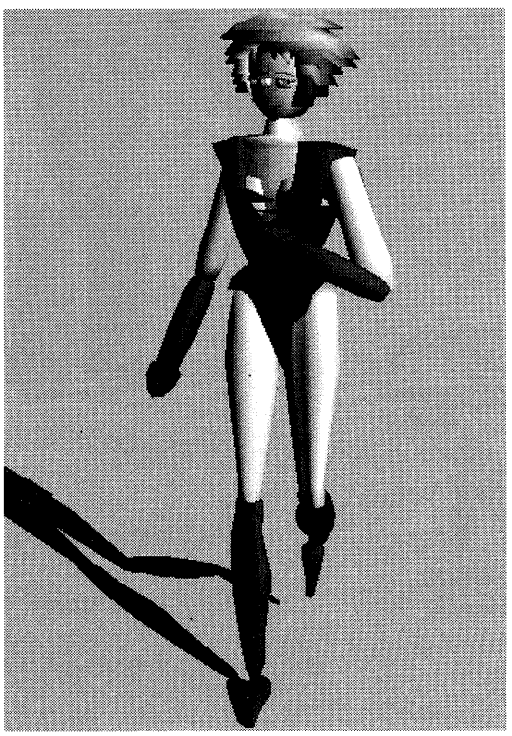
図６　ＣＧキャラクタのフラットシェーディング画像

４．　三次元多関節構造体生成結果

本報告で提案した三次元形状生成手法を用いて、二次元ドローデータから三次元多関節構造体を自動生成した。生成例を図７および図８に示す。



（a）二次元ドローデータ



（b）スムーズ
シェーディング画像

図７　本手法の適用例（１）

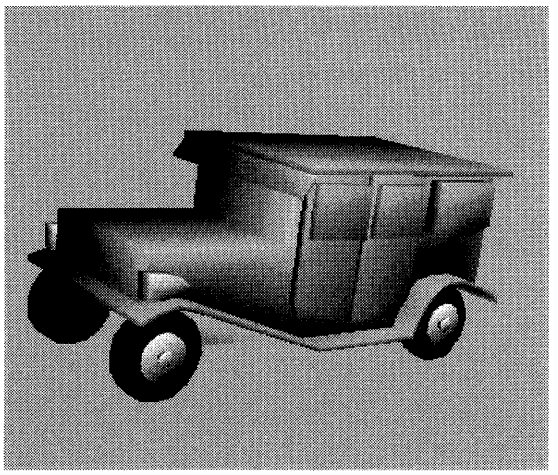
```
#Classic Car
%Scale 2.5 Color 0.6 0.8 0.6 Shape default -15 15

$Color 0.8 0.4 0.4
Shape default -60 60
$Color 0.8 0.4 0.4
Shape default -45 45
$Color 0.8 0.4 0.4
Shape -s default 45 50
$Color 0.8 0.4 0.4
Shape -s default 45 55
$Color 0.8 0.8 0.4
Shape -s default 45 55
$Color 0.8 0.4 0.4
Shape default -40 40
$Color 0.1 0.1 0.1
Shape -cd default -5 5
Offset 45
$Color 0.8 0.8 0.8
Shape default 50 52
$Color 0.8 0.8 0.8
Shape default 52 54

car
rearTire@car
foreTire@car

#Designed by yo1 Horry
1995©HITACHI Ltd.,Tokyo Japan.
```

（a）二次元ドローデータ



（b）スムーズシェーディング画像

図８　本手法の適用例（２）

図７（ａ）は、女性のＣＧキャラクタを表わす二次元ドローデータである。図７（ｂ）は、本手法を用いて図７（ａ）から三次元多関節構造体を自動生成し、スムーズシェーディングによってレンダリングした画像である。ＣＧキャラクタの手足は、自動的に求められた関節位置を中心として、自由に動かすことができる。

図８（ａ）は、クラッシックカーを表わす二次元ドローデータである。図８（ｂ）は、図８（ａ）から自動生成した三次元多関節構造体を、スムーズシェーディングによってレンダリングした画像である。二次元ドローデータの中の文字列を用いた定義によって、ドアやタイヤは奥行き方向に対称に生成されている。各々のタイヤは独立して動かせるようになっている。

５．　おわりに

本報告では、１枚の二次元図形で表現された情報を基にして、三次元多関節構造体の形状と階層構造を自動生成する手法を提案した。二次元図形に与える様々な色や厚みは、図形と対応関係を持つ文字列で定義した。立体化した図形を、図形の傾きを考慮したローカル座標系を用いて階層構造化した。本手法を用いて三次元多関節構造体を生成した結果、本手法が三次元モデリングの一手段として有効であることが確認できた。

謝辞

実験を進めるにあたり御協力を賜った(株)日立製作所 中央研究所の 西岡 大祐 研究員、芦沢 実 研究員に感謝の意を表します。

参考文献

[1]小川：インターネット上にVR空間を構築するVRML；日経CG '95年11月号, pp. 96-113（1995）.

[2]中嶋 他：コンピュータグラフィックス　技術系CG標準テキストブック；財団法人画像情報教育振興協会（1995）.

[3]山田 他：制約充足に基づく三面図理解システム；情報処理学会 グラフィクスとCAD 77-3, pp. 11-18（1995）.

[4]L. Williams：3D Paint；Computer Graphics, Vol. 24, No. 2, pp. 225-233（1990）.

[5]福嶋 他：言語的指示も併用した３次元モデル作成法；情報処理学会 グラフィクスとCAD 77-5, pp. 27-34（1995）.