

カテゴリ汎化特性格の継承階層化データ節約構造 DBをもつ大規模個体群の3DCGアニメーション 通信の検証

正会員 亀田 通[†] , 井手口 哲夫^{††} , 中谷 直司^{†††} , 厚井 裕司^{†††}

3D-CG キャラクタ・アニメーションの通信はデータ量が多く、さらに大規模な個体群のアニメーション表示などでは膨大なデータストリームとなってしまふ。その為、低速な回線などでは通信が困難である。本研究は、これらを回避するため、ユーザ間の通信ではグローバル座標位置などの概略データを送るだけで済むようにし、表示のためのデータは、あらかじめ端末装置側に蓄積して置く方式としている。またこれらのデータは、カテゴリの汎化性で分類し多重継承する階層化した部分派生データのDBとして構成しておき、各端末での不足分は適宜データサーバから配布される。そして各端末がアニメーション表示をする時は、必要なデータを自分自身のデータベースから必要なデータだけを取り出し合成する事でデータを完成させ使用するという方法とする事で、データの分散配置を実現し通信量の節減を実現している。こうする事で大規模な3D個体群でも、それぞれ別個のアニメーション動作を表示をさせることが出来、その応用として同一3D仮想空間内に盛り込んだ多重世界のそれぞれの住民にも、各々の個性を持たせた別個のアニメーション動作表示ができる事を確認した。

Inspection of 3D-CG huge population character animation communication with inherited structured frugal database through narrow bandwidth medium

TOHORU KAMEDA[†] , TETSUO IDEGUCHI^{††} , NAOSI NAKAYA^{†††} and YUUJI KOU^{†††}

Abstract Communication of huge population 3D-CG character animation though narrow bandwidth medium is very hard due to explosive data stream. This paper proposes and inspect the method of outline code communication by global position data among user terminals. Stocked data on special data server is inherited multi layered object oriented database with partial derivation data depend on category generalization. Data for characterization a terminal and database for 3D character animation are delivered from special data server to each terminals in beforehand. When animation display process is exceeding, the terminal pull out the data from own stocks, and synthesize with needed parts. As a result, Frugality of data volume had accomplished by distributed arrangement. In this way, it is confirmed huge population 3D-CG each separated character animation was done and being able to communicate among each inhabitants on each multiple worlds.

1. はじめに

通信のコンテンツも、文字から静止画、音楽・映像配信、さらに立体視映像の配信へと進展し続けている。そしてコンテンツをより現実近づけるためには3DCGが適している。3DCGの仮想空間内での通信に

において、3DCGキャラクタのアニメーション動作のキーフレームデータのカプセル化¹⁾や、キャラクタの多関節の構造操作の研究^{2), 3)}があり、またキーフレームを制御スクリプトとして同期制御して並列処理表示させる研究⁴⁾もある。一般に、いままでは動作キーフレームデータ自体を削減したり、差分化してデータ圧縮する研究が多かった。

筆者^{6), 7), 8), 9)}らも3D仮想物体の曲面の圧縮や3DCGアニメーションの圧縮についての研究を進めている。

いま無限遠の広大な3D仮想空間内に存在する大規

[†]三菱電機 情報技術総合研究所 マルチメディアネットワーク部
Information Technology R&D Center, Mitsubishi Electric Corporation

^{††}愛知県立大学 情報科学部
Faculty of Information Science and Technology, Aichi Prefectural University

^{†††}岩手大学 工学部 情報システム学科
Department of Computer and Information Science Faculty of Engineering, Iwate University

模個数の 3D 個体群を提示する場合、従来の方式では、これを提示するのに必要な膨大なデータを伝送する事ができず、それゆえリアルタイム表示にも、全く対応できなかった。

例えば、キャラクタ 1 体の振付動作（アニメーション）を 1 動作 10 秒として、10 秒間を表示するだけでも、それぞれ 0.1 秒ごとのフレーム毎として計 100 枚分のデータが必要となり、その 1 枚のフレームには、1 体が所有する多関節すべての 3D 位置座標や 3D 方向、3D 速度のデータが必要となる。

また 1 つの 3D 仮想空間舞台に登場する 3DCG 人物の人数が多人数の場合、そのデータは、さらにすべての人数分の量となってしまう。

そして動作アニメーションをリアルタイムに伝送する場合、たとえその伝送路が高速であったとしても、毎回ごと類似なストリーム・データを繰り返し伝送する事となり、このような方式は、やはり通信資産の無駄使いと云わざるを得ない。

しかし、一般にアニメーションは、一連の単位動作の連結で全体動作が構成される場合が多い。したがって各単位動作を別々に転送し、受信側で、その後連結し、全体動作として再構成することが可能である。

本方式は、これらの膨大なデータをリアルタイムに送る事なしでも、つまり低速の通信回線などでも、群集の 3DCG アニメーションを生成し、通信できる事を実現する方法である。

本論文は、生物骨格とその動作アニメーションをオブジェクト指向型データベースとして蓄積し、これらを利用して通信する方法で、そのプロトタイプを製作し、有効性を検証したものである。

以下、2 章では本検証で用いた概略コードによる通信の基本概念と、オブジェクト指向データベースを示し、3 章では DB 構築の詳細を述べ、4 章で実験結果の考察を述べ、5 章では むすび とする。

2. 通信方式の原理

本通信方式は、あらかじめデータを端末側の配布蓄積させておき、場面に必要な概略コードのみを転送し、その蓄積されていたデータを使って端末装置自身で再現し、要素データのみでの通信をするという通信方式をとる。

2.1 3DCG 仮想空間での通信の基本概念

従来の 3DCG 仮想空間通信は、窓型の表示端末を使っているため、その視野以外の事物の表示の必要性がなく、これらの為のデータの通信を省略できた。しかし全天・全地周の 3D 視認の出来る CABIN^{*1}システムやヘッドマウントディスプレイ (HMD)^{*2}などでは、全視野となり、またユーザ個々の見る方向が異なるため、仮想空間内に存在する全個体をあらかじめ送付して置かねばならない。

図 1 は、全天・地周型の 3D 視認装置を使った仮想空間内での通信の概念で、この場合、ユーザは他のユーザか、もしくはデータサーバシステムが作り出した舞台とだけしか通信していないことを示している。それゆえ、閉じた通信ネットワーク内では、これだけ

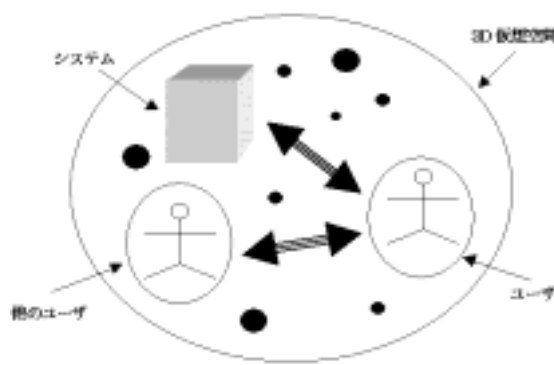


図 1 全天・全地周型の視認端末での仮想空間での通信の概念

Fig.1 Concept of outline code communication
の通信回線を確保するだけでよい。

図 2 は、本論文でのデータサーバを中心とした 3DCG 仮想空間の通信の構成を示したもので、各種データを供給する為のデータサーバを中心に、各ユーザ

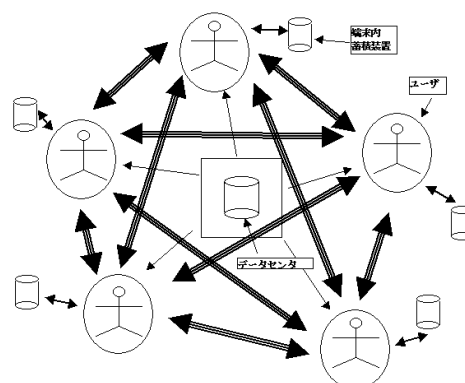


図 2 データサーバを介しての DB アシストのマルチユーザの仮想空間内通信の概念

Fig.2 Concept of virtual space communication assisted by database

*1 <http://www.iml.u-tokyo.ac.jp/cabicl/wholesystem.html>

*2 http://www.zdnet.co.jp/news/0209/10/njbt_04.html

が操作する全周囲型の 3DCG 表示端末が配置されている。こうすることで、その場面の群集の 3D アニメーション表示を各端末で独立の行うことが可能となり、さらにこのことから 1 個の仮想空間内に、多数の世界が共存できる多重世界の表示が個別に可能となる。

図 3 は、これらの表示の例を示したもので、大規模個体群の群集であっても、それぞれが個別動作をする表示が可能であることを示している。

図 4 は、これをさらに展開した例で、1 個の仮想空間に多数の異なる世界を盛り込んだ例である。

図では理解しやすいように建物のような表現したが、本当は、同一の 3D 空間に多重に存在する。

現在では、通信コンテンツとして文字、画像、映像、立体視映像、音楽、3DCG、3DCG のアニメーション、VR や、またそれらのメタデータなど、ありとあらゆるデータが、混在するようになった。それゆえ、これ

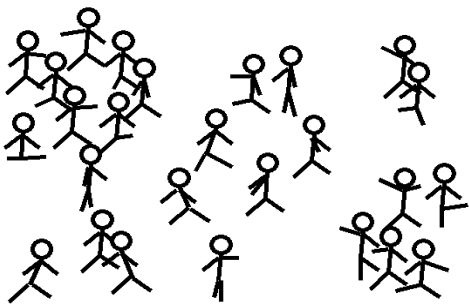


図 3 大規模個体群の表示形式
Fig.3 Huge population character animation

らの種々雑多なデータのすべての形式を扱うには、もはや従来の考え方のデータベースでは、対応できない。

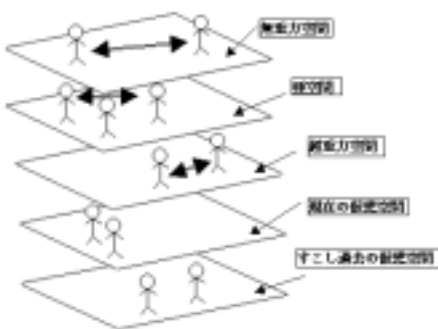


図 4 3D 仮想空間の多重世界構造
Fig 4 Multiple worlds in unit virtual space

それゆえ本論文では、3DCG アニメーション以外にも対応できるデータベースとして、オブジェクト指向型データベース (OODB) を採用した。

2.3 概略コードでの通信

概略コードを使った通信方式での処理について図 5 で説明図する。

この概略コードは、2 種類のデータ群から構成されている。1 つは個体の属性に関する概略パラメータで、もう 1 つは、仮想空間全体での座標系に関する概略パラメータである。その処理の過程も図 5 で説明する。

まず受信したデータは、個性データと全体座標系データとに弁別される。個性データは、生物タイプコードと、骨格数、骨格タイプコード、振付コード、化身コードで構成される。また全体座標系データは、世界コード、その世界でのグローバル位置座標 $X_i Y_i Z_i$ 、初期向き方向、その場の回遊データの群から構成されている。

個性データの処理としては、選定された生物タイプコードの範囲内の骨格コードを取得し、その骨格数の

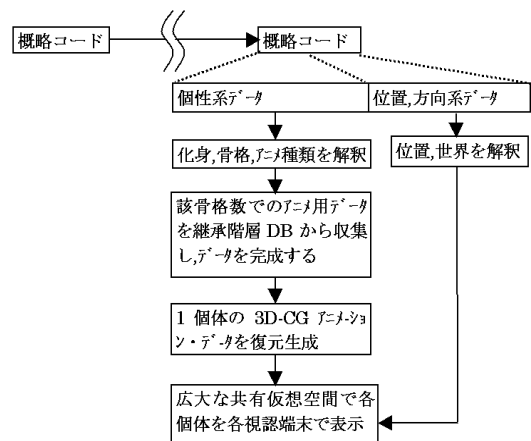


図 5 3DCG アニメの概略コード受信処理図
Fig.5 Figure of outline code receive system

範囲内の動作データ群を検索し取得するようにしている。化身選択コードはポリゴンの種類を選択する。これらすべての処理はパターンマッチング方式で処理している。

一方、グローバル位置座標や、世界選定コードは、場所を指示するデータなので、無処理のまま残され、アニメーション表示の準備がすべて完了してから、指定の場所に表示するのに利用される。

3D 仮想キャラクタ物体の動作振舞データは、曲線生成用の制御点のデータであり、B スプライン曲線生成処理で生成された補間データの軌道データでアニメーション動作が復元される。

3 . カテゴリ汎化特性格の継承階層 DB と その処理

3.1 特徴と利点

本方式での利点は、パッケージ化されたアニメーション動作の骨子を送付する事で、データ送付の容易性を実現させた事と、蓄積されていたオブジェクト指向型データベースを使用するとき、その継承部分の変化分をオーバーライドすることで、場面のインタラクション要求に応じて、即時的に変化させて利用することが出来る点である。また目的の動作オブジェクトを生成する時、分散配置されている部品群から必要な部品を収集し、組み上げて、要求に応じた動作オブジェクトを完成させる事で、柔軟な対応ができる。また構成している部品から見れば、色々なオブジェクトから多数回引用され利用される事で、有限の蓄積データ資産を見掛上、無限個近くあるように、有効に利用される点にある。

3.2 カテゴリ汎化特性格の階層構造

図6は、本実験で用いたデータの1例である。人間の骨格の基本を設定し、これを姿勢変形していくことで、生物タイプと動作タイプを誘導する仕組みを作っている。

図7は、これら骨格データを実際に蓄積する方法で、腕の長さや角度を含んだ直線表現の本体データ列を、マルチリンクのツリー構造にしている。各データの前面には、このデータ列ID番号を配置し、後面に

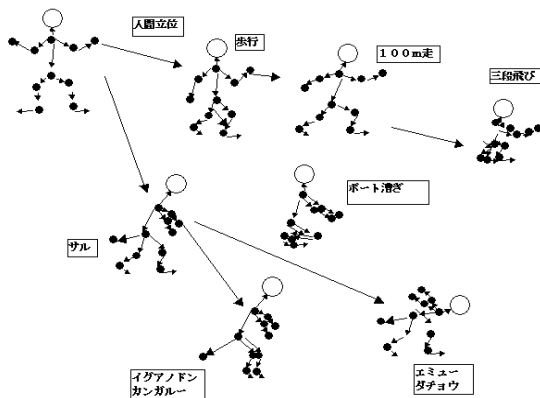


図6 骨格構造データの階層の様子
Fig.6 Method of hyperstructured data derivation with humanoid skelton.

マルチリンク用の可変長のリンク情報を付加する構造としている。

3.3 部分派生データの収集成成処理

図8は、データの汎化特性に応じて階層化されたデータの構造の設計図の1例であり、UML v1.2 記法で

表示した。

これらは、オブジェクト指向データベースの各クラスに付属する。またクラス階層でのインヘリタンスは、他の視点からのポリモーフィズム性も持たせている。

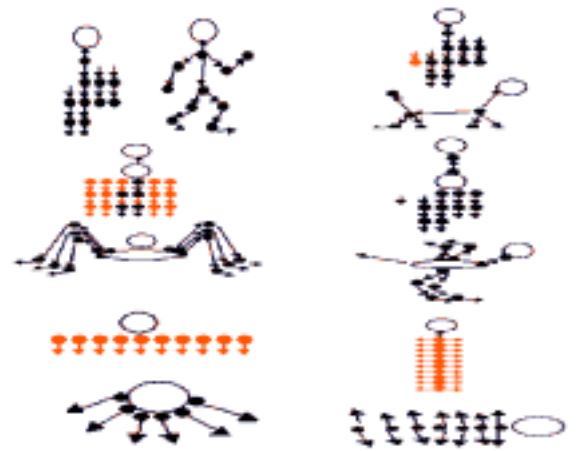


図7 骨格データの蓄積方法
Fig.7 Storage method of born structure data

また、インタラクティブ対応時に即時的に変化する必要がある事から、この階層の相応部分をオーバーライドさせ、このインヘリタンスを変更する。

3.4 動作単位での付随データ

付随データは、オブジェクト指向データベースの各

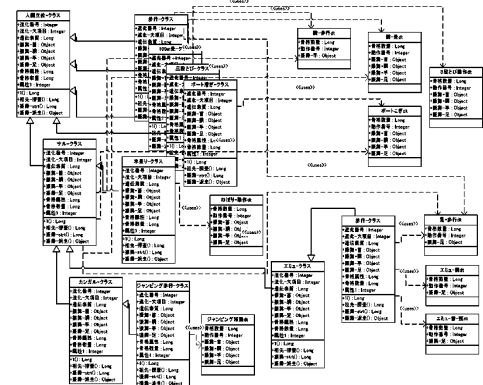


図8 UML v1.2 表記での3D-CGアニメーション動作データの構造例
Fig.8 Example of object oriented structured database for animation

クラスに付属しており、以下のパラメータである。

- 1) B-Spline 制御点 + 区間
- 2) 軌道上速度

1)は、スプライン閉曲線の制御点パラメータで、現在は5フレーム分を保有しており、3D-CGへの内訳は、フレーム座標として、方向ベクトル6個、位置ベクトル3個で構成されている。

2) の速度は、制御点間の区間の各速度を補間点の個数の逆数として制御されている。軌跡曲線は、N + 1 個のパラメータのあとに、その動作形式（閉ループ連続巡回、開ループ 1 回のみ動作）などである。

3.5 アニメーションの例

本提案でのオブジェクト指向型データベースは、アニメーション動作用に特化している。

上位の「骨格」のクラスから、下位の「散歩」クラスまで、階層化される。そして、各クラスの属性は下位クラスへと継承される。

また、新しく送付されたデータ群から、新しくクラスが派生する様子を、「歩行」クラスからの新サブクラスとして生成していることを図示する。

図 9 は、本論文で用いた「歩行」の振付データを使用し、歩行動作のキーフレーム 5 枚分から連続したフレームを生成し、表示した例である。

図 10 は、1 個の仮想空間に多重世界を盛り込んで、各世界の各住民の様子を表示したもので、この場合、他の世界の住民は、色味を帯びた半透明な物として表現している。

3.6 概略コード通信の検証

通信実験は、Windows 2000 Pro[®]OS の PC で、サーバとクライアントの両方のプロセスを同時に動作させ、1 台のパソコン内だけでプロセス間 TCP / IP 通信実験で行なった。

図 11 は、この時のサーバプロセスとクライアント

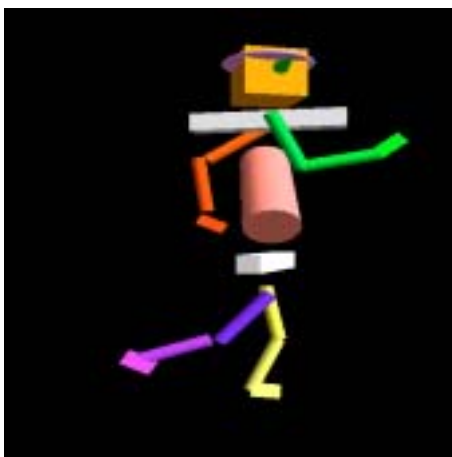


図 9 3DCG キャラクタ 1 体のアニメーション表示例
Fig. 9 Sample of 3DCG one character animation.

プロセスを同時に動作させている CRT 画面の図である。

プログラム開発環境は、同 Windows 2000 Pro[®]OS の .NET Framework[®]Ver 1.0 の下で、VB.NET[®] 言

語で行い、DirectX 7.0a[®]の Direct-3D の 3D-CG グラ



図 10 多重世界機構内での半透明な生物の 3DCG アニメーション

Fig.10 Half transparent 3D-CG undefined creature animation in multilayerd world

フィックライブラリを用いた。また TCP/IP 通信では、同 Framework の COM+1.5 オブジェクトを利用した。

3.7 アニメーション単位動作の復元

本検証では、B-スプラインの閉ループ曲線生成をしている。関連づけて抽出された軌道生成用の制御点データは、Direct3D の Animation2 ライブラリでの interpolation メソッドにより、補間データとして充填され復元される。また動作速度も補間データ量で設定される。



図 11 同一 PC 内でマルチスレッドでクライアントプロセスとサーバプロセス間の概略コード通信している様子の画面

Fig.11 Client-Server outline code communication for 3DCG Animation (within same PC)

4 考察

4.1 理論的考察

これらカテゴリの汎化性に応じて階層化した派生値の DB は、連分数的な式で表現されると考える。(付録 1 参照) また階層形態は図 12 に示した例のごとく、3 種類の大別される。これらを式 (1), (2), (3) に示した。そしてこれらを統合して一般化した式を (4) に示す。節約効率は付録 2 で考察した。従来方式では式 (5) であるが、本方式では式 (6) の節約構造となり、それゆえ、結果的に式 (7) となると考えられる。

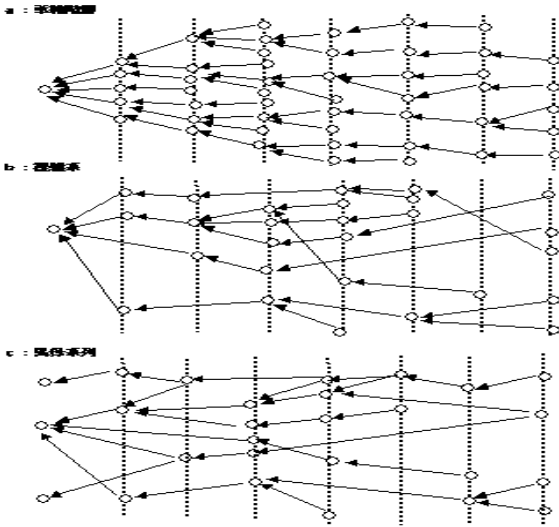


図 12 DB の階層構造の 3 大別の模式図

Fig.12 Model of 3 Type DB structure

4.2 節約率の考察

いま具体的な数値を代入してみると、

動物種類 = 50 種
表示個体群数 = 10000 個
初期姿勢数 = 20 座位
関節数 = 30 関節

とし、実際に作成した DB でのデータ階層の様子をもとにモデルを作成し、これらから予想してみるとベリーダンスやフラダンスのように、手首などの末端に向かうほど微妙な動作を必要とするので、関節の末端にいくほど $2^{1.2}$ に増加していくと仮定して、従来方式では、

$$\begin{aligned} \text{データ数} &= 50 * 10^{4 * (N_0 * 20 + N_1 * 2^{1.2} + \dots)} \\ &= 544,144,735 \text{ 個} \end{aligned}$$

のデータの個数が必要となる。

しかし、表示する群の個体数に無関係であることから、本方式の節約の方式では、

$$\text{データ数} = 2 * (1 + 2 + \dots + 30)^2$$

$$= 432,450 \text{ 個}$$

となる。

これは、かなり少ないデータ個数で、この例によると、約 $1/1258$ にデータ量が節約された事となる。そして、これはまた、表示しようとする群の個数が増せば増す程、その効果が顕著に表れてくるようになる。また、アニメフレーム数/秒や、関節の原点座標 Xyz や、当初の方向、などの、アニメーション動作を再生する為の細かなデータが含まれているので、これらは、階層構造データを倍化するものと考えられる。

4.3 本実験で生成したコード量

各サブルーチンのコード量は以下にしめすもので、処理時間の把握できる。

: クライアント・サーバ通信 = 19,988 バイト
: オブジェクト指向 DB 制御 = 29,446 バイト
: 3DCG グラフィック = 2,298,986 バイト

処理する数値は、単精度の浮動小数点で計算した。

仮想空間の開催されている舞台ネットワークに任意のユーザ端末が承認され、新規に入場したり、また突然、退場したりして、3D 個体が突然出現したり、消滅したりして、舞台の登場人物の数がいつでも変化する環境にも対応できる特徴がある。

つまりゲームなどでは、あらかじめストーリーに埋め込まれたシーケンスに応じて、登場人物が、出現・消滅しているが、本方式では、全くストーリー性がなく、それでも対処できる事が特徴である。

なお本 3D グラフィックには、v70a 版のライブラリを用いたゆえ、人物の身体部分をグローバル座標系として、足などの多数個の構成部分をそれぞれ相応のローカル座標系を生成してから、これに構成した。そのため多数のローカル座標系を生成させる必要があり、またそのそれぞれに、移動や回転などの細かいパラメータを設定した。

しかし、最近の v8.0 や v9.0 版のライブラリを用いれば、多数個のローカル座標系を生成する必要もなくなり、関節部分のポリゴンのやぶれの縫合が可能となり、より容易にプログラム開発ができるものと考えられる。

5. むすび

本論文では、古代(| オルドビス紀 | カンブリア紀 | ベルム紀 | 石炭紀 | デボン紀 | シルル紀 | 白亜紀 | ジュラ紀 | 三畳紀 | 第四紀 | 第三紀 |)などの

所謂、遺伝子爆発の時代での生物進化を参考に、骨格やアニメーション動作データを範疇別階層構造データとして構成した。このような方法を導入することで、アニメーション動作を、それぞれ独立に動作表示させる事ができた。また概略コードのみで通信する方式を実現するため、動作データはオブジェクト階層型 DBとして各端末側に蓄積させた。ただDBの性質上、フォワード・キネマティクスであった。

今後、インバース・キネマティクスのDBの構築や、機能の追加を考える。また最新バージョンのグラフィックエンジンの利用によるポリゴンの自動縫合も考える。

参考文献

1)平松,八村:身体動作の記述と表示のための実行制御環境の開発,情報処理学会論文誌 Vol.40, No.3,pp.939-948 (1999).
 2)中村,山根,永嶋:構造変化を伴うリンク系の動力学計算法とヒューマンフィギュアの運動計算,日本ロボット学会誌, Vol.16, No.8, pp.124-131 (1998).
 3)Tsutsuguchi, K., et al: Human walking animation based on foot reaction force in the three dimensional virtual world, The Journal of Visualization and Computer Animation, Vol.11, pp.3-16 (2000).
 4)吉川,正,瀧:実時間三次元アニメーションシステムの

並列オブジェクトモデルを用いた実現手法,情報処理学会論文誌, Vol.40, No.5, pp.2183-2191(1999).

5)P.K.Andlegh&M.R.Gretzinger,楠 訳:オブジェクト指向データ・システム設計 - 分散環境における - , (株)トッパン ISBN 4-8101-8581-86)亀田,井手口,厚井:再帰的NURBS処理手法を用いたモバイル通信での3次元仮想空間内での自由曲面の圧縮通信方法の検討,情報処理学会, Vol.99, No.13, 第8回モバイルコンピューティング研究会, 99-MBL-8 pp.55-62 (1999).
 7)亀田,井手口,厚井:仮想3次元NURBS自由曲面の制御点などのデータ削減と補償的最適化手法の提案,情報処理学会, 第10回モバイルコンピューティング研究会, 99-MBL-10 Vol.99, No.13, pp.1-8 (1999).
 8)亀田,井手口,厚井:ポリゴン3次元仮想物体の節減NURBS曲線圧縮転送方式の提案,情報処理学会, Vol.42, No.10, pp2415-2422 (2001).
 9)Kameda, T., et al: Proposal of NURBS curve forming with Genetic Algorithm and pseudo newton method on the polygon object as particle group, Proceedings of the ACIS 2nd International Conference, pp1018-1025 (2001).

付 録

付録1: 範疇の汎化別階層化データ構造の模型

a: 単純 階層構造

$$Total_A = A + A' + A'' + A''' + \dots \quad \text{----- (1)}$$

$$where \quad A' = \frac{1}{\alpha_1} * A \quad A'' = \frac{1}{\alpha_2} * A' \quad A''' = \frac{1}{\alpha_3} * A'' \quad \dots$$

$$= A * (1 + \frac{1}{\alpha_1} + \frac{1}{\alpha_1 * \alpha_2} + \frac{1}{\alpha_1 * \alpha_2 * \alpha_3} + \dots)$$

b: 複雑系 階層構造

$$Total_A = A(1 + \frac{1}{\alpha_1} + \frac{1}{\alpha_1 * \alpha_2} + \frac{1}{\alpha_1 * \alpha_2 * \alpha_3} + \dots) \quad \text{----- (2)}$$

$$+ A(\frac{1}{\alpha_3} + \frac{1}{\alpha_1 * \alpha_4} + \frac{1}{\alpha_2 * \alpha_5 * \alpha_6} + \dots)$$

c : 異親系から混合継承の階層系列

Total_ABC =

$$\begin{aligned}
 & A * (1 + \frac{1}{\alpha_1} + \frac{1}{\alpha_1 * \alpha_2} + \frac{1}{\alpha_1 * \alpha_2 * \alpha_3} + \dots) + B * (1 + \frac{1}{\beta_1} + \frac{1}{\beta_1 * \beta_2} + \frac{1}{\beta_1 * \beta_2 * \beta_3} + \dots) + C * (\dots) \\
 & + A' * C' * (1 + \frac{1}{\alpha_1 * \gamma_1} + \frac{1}{\alpha_1 * \alpha_2 * \gamma_1 * \gamma_2} + \frac{1}{\alpha_1 * \alpha_2 * \alpha_3 * \gamma_1 * \gamma_2 * \gamma_3} + \dots) + A' * B' * (\dots) + \dots \\
 & + A' * B' * C' * (1 + \frac{1}{\alpha_1 * \beta_1 * \gamma_1} + \frac{1}{\alpha_1 * \alpha_2 * \beta_1 * \beta_2 * \gamma_1 * \gamma_2} + \dots) \quad \text{----- (3)}
 \end{aligned}$$

親をそれぞれA, B, Cとし, 各層のN個のカテゴリにM個のデータがあるとする, これらを統合して,

Total_DB =

$$\begin{aligned}
 & A(1 + \sum_{n=1}^N \frac{1}{\sum_{m=1}^{N,M} (N_{NM} * \prod_{m=1}^M \alpha_m)}) + B(1 + \sum_{n=1}^N \frac{1}{\sum_{m=1}^{N,M} (N_{NM} * \prod_{m=1}^M \beta_m)}) + \dots \\
 & + A'B'(1 + \sum_{n=1}^N \frac{1}{\sum_{m=1}^{N,M} (N_{NM} * M_{NM} * \prod_{m=1}^M \alpha_m \beta_m)}) + B'C'(1 + \sum_{n=1}^N \frac{1}{\sum_{m=1}^{N,M} (N_{NM} * M_{NM} * \prod_{m=1}^M \beta_m \gamma_m)}) + \dots \\
 & + A'B'C'(1 + \sum_{n=1}^N \frac{1}{\sum_{m=1}^{N,M,L} (N_{NML} * M_{NML} * L_{NML} * \prod_{l=1}^L \alpha_l \beta_l \gamma_l)}) + \dots \quad \text{----- (4)}
 \end{aligned}$$

となる.

付録2 : 従来と本方式のデータ量の比較

従来の方式では, そのトータルのデータベースのデータ量は,

Total_DB =

$$\text{動物種類} * \text{表示個体群数} (N_0 * \text{座位数} + N_1 * \text{関節 1} + N_2 * \text{関節 2} + \dots + N_N * \text{関節 N}) \quad \text{----- (5)}$$

となり, その表示個体数に比例する.

しかし, 本提案の方式では, 有理数の連分数展開での法則から,

いま

$$2N \geq N(1 + \sum_{n=1}^N \frac{1}{\prod_{m=1}^M \mathfrak{R}_{\alpha\beta}}) \quad \text{----- (6)}$$

であるから, これをもとの表現式に適用すると先の不等式を適用して

$$2N(A+B+C+\dots)^2 \leq \text{Total_DB} \quad \text{----- (7)}$$

となる.