

3DCG を簡単に使える方法の検討

柳瀬 龍郎[†] 中山 大輔[†] 田村 信介[†]

[†] 福井大学大学院 工学研究科 情報・メディア工学専攻
〒910-8507 福井県福井市文京3丁目-9-1

要旨

汎用アプリケーションにおいては、文字や数値の出力は `printf()`、入力なら `scanf()` 等の入出力 API を用いて簡単に処理の記述を行う事ができる。本研究は情報の表示を 3DCG にて容易に行う事が出来るような枠組みについて検討する。その具体的な方法として、入力処理は簡潔な記述により、GUI を用いた方法を可能とし、また 3DCG 出力も簡単な記述により可能とするシステムの提案を行う。実装するためには GUI 入力のための API とプリプロセッサを構築する必要がある。プリプロセッサによりソースコードに用いられた API を検出し、その引数等から CG 出力画面と GUI 構成のための部品情報を抽出する。アプリケーション実行時にこれらの情報を用いて GUI を構成し、CG オブジェクトの表示と、GUI 部品を介した数値等の入力処理をおこなう。

Examination of a method to easily use 3DCG GUI

Tatsuro Yanase, Daisuke Nakayama and Sinsuke Tamura

Fukui University

Abstract

In a general-purpose application, one can easily write a letter and the numerical output and the input by `print()` and `scanf()`. This study examines the method that can output indication of 3DCG easily. In addition, the reports propose an I/O system architecture to use 3DCG for output of processing, and a GUI using mouse to input numerical parameters for application program. One can use an input method and the 3DCG output using GUI by simple statements. For the implementation, we need a preprocessor and an API for the GUI input. A preprocessor detects API used in a source code, and extract parts information for a 3DCG output and GUI construction from the arguments. A GUI will use this information at the time of application execution and constitute a GUI and display CG objects and input handling of numerical value through GUI parts.

1. はじめに

ゲームやコンピュータグラフィックによるアニメなど等のコンシューマ向けグラフィックスに関するハードウェア技術やそのための専用ツールソフト類は極めて高度に発達しまた市販され利用に供さ

れている。だがこれらのハードウェアやツールソフト類はあくまでいわばグラフィックのためのグラフィックツールである。

アプリケーション制作者が処理を対話的に行うためにもちいることができる簡

単な記述にはC言語ならば `printf()` や `scanf()` といったライブラリ関数を呼出すしか手段がないのが現状である。きわめて原始的であるとさえ言えるような、これらの手段は「対話的」処理をするにも不可能に近い機能でしかない。とはいえ、OpenGL や VRML 等の 3D シーン記述言語や Java3D の API を用いることにより、グラフィカルな出力や入力を行う事も可能ではある。しかし例えば OpenGL により 3 DCG 出力を行うには単純な API の利用で関数の呼出を行うだけでは済ますことは出来ない。初期設定や様々な変数やパラメータの設定。なにより OpenGL 特有のイベント型プログラミングの記述とイベント処理におけるプログラムエラーのデバッグ困難性等の理由により、提唱されてからかなりの歴史があるにもかかわらず、一般のアプリケーションにおいての出力手段としてはそれほど利用されているとは言えないのが現状である。

既存の 3DCG 表現手段について考てみる。PATRAS は画像生成処理の並列化を目的として大規模な並列システムにおいて利用されている[1][2]。また、あまた市販されているグラフィック生成ソフトウェア[3][4][5]等はそのほとんどは 3D アニメーションやビジュアルエフェクトを作成する 3DCG ツールであり、モデリング、アニメ、レンダリングなどが行える 3DCG ツールである。しかし汎用アプリケーションの簡便な 3 DCG 出力システムとして利用するための機能は全く備えていない。RVSLIB[6]と PATRAS はリアルタイム可視化画像生成システムであり、アプリケーションコードに可視化のための API コードを挿入することにより画像生成が可能となっている。また、可視化のためのパラメータを実行中に変更することも可能である。また、広く市販に供されている可視化ソフトウェアとして AVS[7]があげられる。AVS は何らかの方法にて用意されたデータを可視化するためのツールであり、基本的にはいわゆる「オフライン」で使用するパッケージソフトであるが、最近では AVS でもリアルタイム表示の機能を備えるように改良さ

れているようである。

これらの様々なソフトウェアは、いずれも「お絵かきのための道具」または「電子計算機による処理」ではあるが「バッチ処理計算」による「データの取得」を前提にした基本的スタンスが基底に置かれている気がするのである。すなわち、「対話型処理」(あるいは「TSS 処理(?)」)なんとまあ古い響きであろうか!)のための CG 出力と GUI を利用したマウスによる入力システムとしての手段、という概念が欠如しているようにも思える。シンプルではあるが“scanf”や“printf”の使いやすさは、これら上記のソフトウェアとはかけ離れた対話的処理の極めて有効な手段である。これらの二つの関数は残念ながら文字列と数値のみの入出力が可能である。同様な単純さで 3 DCG 出力とマウスによる数値等の入力可能なシステムは未だ用意されていないのが現状である。

2. 研究の目的

本研究は、上記の問題に対する一つの解決策としてアプリケーションにおいて 3 DCG 出力を単純に関数の呼出で可能とし、また数値や ON/OFF などの値の獲得を一つの関数の呼出しのみにて GUI 部品などにより可能にするシステムを提案する。

2.1 期待される結果

システムが実装されれば、3 DCG 出力と GUI からの入力のための API を用いてアプリケーションを記述すれば、プリプロセッサとコンパイラを通すだけでオブジェクト実行の用意ができる。その後 GUI の起動コマンドを実行するだけでこれら API を用いたアプリケーションの実行が開始され、アプリケーションにおいて API で行われた 3 DCG 出力が実行され、また GUI によるパラメータ数値のアプリケーションでの獲得が入力 API の実行において簡単にマウスで行う事ができる。このことは、C 言語等を用いた、従来の手続き型プログラミングスタイルで 3 DCG の出力が出来、従来の“scanf()”関数

を用いた数値の獲得のように、容易に GUI を通じて “スライダー” や “ボタン” 等 GUI 部品を利用した数値などのパラメータ獲得がアプリケーションでできることを意味する。従ってアプリケーション制作者はコードの記述にあたって、イベント駆動型の処理に気を使うことなく、プログラム全体を通じて手続き型プログラミングスタイルの一貫性を保つことができる。

API として考えられる例としては：

出力関数 OUT(): 3D オブジェクト属性データ送信

入力関数

- IN_slider(): GUI のスライダの値入力
- IN_button(): GUI のボタンの値入力
- IN_key(): キーの一文字入力, etc
- IN_Init(): 入力部品の初期設定[省略可]

2.2 実現の戦略

3DCG 出力と GUI 入力のための API とプリプロセッサを構築する。ソースコードに用いられた API をプリプロセッサにて検出し、その引数等から CG 出力画面と GUI 構成のための部品情報を抽出する。このとき、不足した情報があれば既定値を用いる。アプリケーション実行時には、これらの情報を用いて GUI を構成し、アプリケーションから出力される CG オブジェクトの表示と、GUI 部品を介した数値等のアプリケーションに対する入力処理をおこなう。

3. 概要設計

図 1. にターゲットシステムの概略を示す。システムはアプリケーション実行部と 3DCG を出力する GUI 部から構成され、互いにネットワークで接続され、GUI 部はサーバ、アプリケーション実行部はクライアントとしてネットワーク結合される。

GUI 部はネットワーク結合のサーバとして起動される。またこの GUI 部によって RPC にてアプリケーション部が起動される。

アプリケーション部はネットワーク的にはクライアントとして動作する。コ

ネクションが成功すればアプリケーションは処理を開始する。

アプリケーションにおいて 3DCG 出力関数(以下においてこれを仮に OUT() とする)が呼出されたときはそのデータを GUI 部に送信し GUI に表示する。

アプリケーションにおいて入力関数(以下においてこれを IN() とする。図 1 の IN_slider() がその API の一つを表す)が呼出されたときは、その API の引数で指定された GUI 部品で入力された値が返値と得られる。

このとき GUI の部品に関するデータは、初期値設定後、部品の状態に変化があることを全てイベントとして GUI 部から計算システム部に情報が送信され、IN() の関数の中で全てこれらの処理と管理が行われており、個々の IN() 関数が呼出されたとき、部品の値のみが返値として返される。

4. システムの動作概略

上記において述べた機能を実現するために必要な全体の処理の流れを図 2 に示す。アプリケーションの製作から実行までの過程は以下のように行われる：

- (1) 本研究の API を含むアプリケーションコード (App) を作成
- (2) 本提案のプリプロセッサ (pp) による処理を行う

[pp. のスキャンにより、コードの中で使用されている 3DCG 出力 API と GUI 部

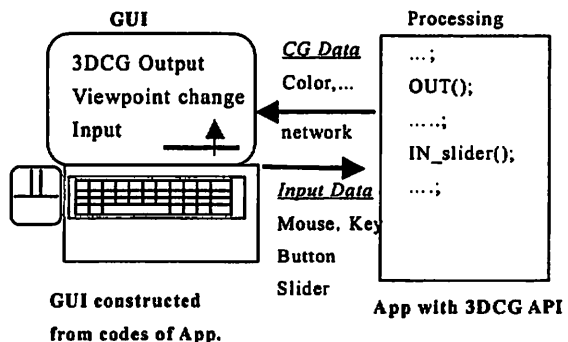


図 1 システムの概略
Fig. 1 System Block Diagram

品による入力 API が検出され、その引数

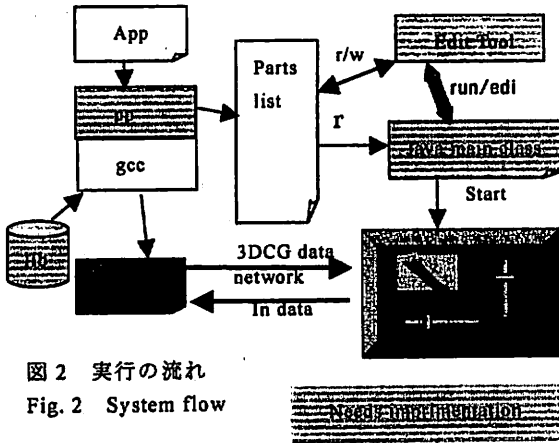


図2 実行の流れ
Fig.2 System flow

等からデフォルトの GUI 構成のための部品情報がリスト (Parts list) として出力される]

(3)コンパイルの後、本研究の API のライブラリと共にリンクしアプリケーションオブジェクト(a.outとする)を得る。

(4)GUI 起動コマンド (Java main) によって GUI 部を起動する。

[GUI が起動されると、Parts list が読み込まれ GUI 全体が部品とともに構成される。さらにその後 RPC にてアプリケーション (a.out) が起動される。そのうち、GUI 部はクライアントとなるアプリケーション部からのソケットの結合要求をサーバとして待機する。RPC で起動されたアプリケーション a.out から GUI 部接続の要求が発行されるので、サーバである GUI 部はこれをアクセプトし、これ以後アプリケーション全体の処理が行われる]

(5)必要であれば、GUI 編集 Tool を用い、直接マウス等を用いて GUI の構成を修正し(4)へ。

[GUI 編集 Tool は GUI 構成部品の全ての属性値 (例えばスライダーであれば配置の位置や方向、あるいは部品そのものの大きさや、最大値/最小値、目盛りの添付等) を編集することが可能。編集結果は全て Parts list に反映される]

(6)実行時においては、アプリケーションは GUI 部品であるスライダーやボ

タンのイベントや値情報をアプリケーション部の IN()関数処理部に送信する。またアプリケーションからの OUT()による 3DCG オブジェクトの座標その他の属性データは、ネットワークを介して GUI 部に送信され 3DCG 表示画面に表示される。このように、アプリケーションの制作においては 3DCG のオブジェクト表示のための諸々の設定、表示画面の構築や視点に関する設定等のコードに関する記述は省略することが出来る。この場合 pp による既定値が用いられる。これは、3DCG オブジェクトはおおよそ原点近辺に座標が配置されるという習慣を利用したものである。

むしろ、これらの既定値は GUI 編集 Tool によって修正可能であることは言うまでもない。

また、GUI 入力部品についても同様であり、アプリケーション制作にあたっては入力 API を用いるだけでよい。

5. 機能の実装設計

5.1 必要な機能

システム全体の実装に必要な処理機能は：

(1)App のソースコードから部品情報のリストを出力するプリプロセッサ(pp.)

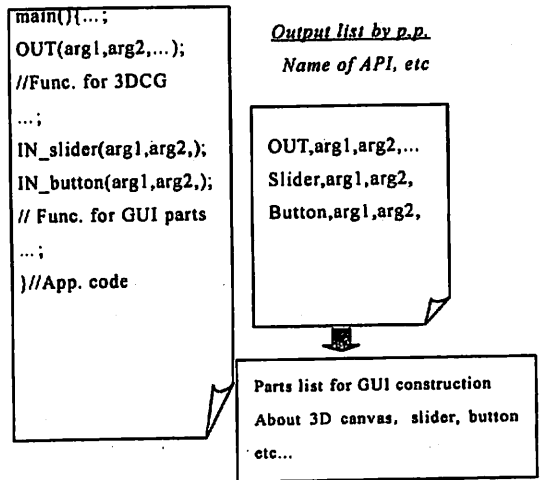


図3 プリプロセッサの出力
Fig.3 Output of preprocessor

(2)デフォルトの 3DCG キャンバスを用意し、部品情報を読んで部品を GUI に用意する (Java) main prog.

(3)(出力)3D キャンバスに CG オブジェクトを表示する API ライブラリ

(4)(入力)GUI 入力部品から入力値を取得する API ライブラリ

(5)GUI 編集 Tool

5.2 プリプロセッサ

アプリケーションをスキャンする役目のプリプロセッサ pp の主な機能は、使用されている API の名前とその引数の抽出。さらに API の機能を実装するために必要な GUI 構成部品の同定と、配置に必要な座標やその他の属性値の既定値を決定し Parts list に書き出すことである。

5.3 GUI 部品情報 (Parts list)

図 3 の左側のアプリケーションコードでは、アプリケーション制作者が printf() や scanf() を用いるのと同様に GUI に対する出力 API OUT() を用いて必要とする 3DCG 出力や、入力 API IN_slider(), IN_button() を用いて GUI 部品のスライダーによって設定された値やボタン部品の ON/OFF の状態値の獲得を実現するための実行ステートメントとしてのコードを記述する。プリプロセッサはこれらを検出し出力するリストに、API の名前や引数、OUT とその引数 arg1, arg2, ... や IN_slider, IN_button とそれら API の引数がリストに全て書き出す。さらにこの他、3DCG 出力 API ならグラフィック出力のための画面サイズや画面の配置位置などの値。また IN_slider API ならスライダー部品の GUI 画面上における配置の位置や初期のレバーの位置。IN_button ならボタン部品の状態の初期値等の各種属性値が既定値として名前や引数と共に Parts list に書き出す。

5.4 GUI 部

GUI 部は実行アプリケーションの内容にかかわらず所定の機能を持つコマンドが用意されなければならない。これはアプリケーションの実行に先立って何時でも動作される必要があり、これは起動プログラムの機能を実装されなければなら

ない。図 2 では Java main がその起動のコマンドとして示されている。

この起動コマンドによって GUI 部が起動されると GUI 部品に関する情報リストが読み込まれ、部品が GUI 画面に配置され GUI 全体がされる。

5.5. 画面編集ツール

しかし、実際には複数の GUI 部品すなわちスライダーやボタンを既定値による配置のまま使用する場合は少ないものと考えられるため、編集する Tool が必要と思われる。

6. API

本 API は GUI とネットワークを介して入力機能と出力機能を実現しなければならない。この目的のために 3DCG の出力関数 OUT() を、また GUI 部品による入力機能すなわち IN_slider() なら GUI 部品であるスライダーの設定値をネットワークを介して受け取る処理等をそれぞれライブラリとして用意する必要がある。まず 3DCG 出力関数 OUT() については、例えば 3 次元空間に三角形の面を表示するなら、各頂点の 3 次元座標と面を塗りつぶす色指定する RGB の値、さらにその面の透明度を与える引数等が必要である。関数処理においてはこれらの値を指定したプロトコルに従って順次、GUI 部とコネクされたネットワークに送信する。次に GUI 部品による入力について、例えばスライダーの設定値を読む関数 IN_slider() の処理においては、関数内の返値は最初の段階で既定値で初期化される。次に、GUI の部品としてマウス等により操作されるとイベントが発生し、これが全て GUI 部からアプリケーション実行部に対して送信される。このため、入力関数の全体で全てのイベントに関する処理部において GUI の状態をイベントにより把握し、その後例えば、スライダー関数 IN_slider() においては、それまでに送られてきた最後のイベントによって与えられたスライダーの値を関数の返値とし、関数処理を脱出する。他の GUI 部品であるボタンの状態を取得する API についても同様である。

7. 実験による予備評価

本提案を実装した予備実験により、サンプルとして用いたアプリケーションコードでは、API の使用により表 1 に示すようなコードの行数の減少が得られた。

表 1 サンプルによるコードの減少

行数	本 API なし	API 使用
表示部分	(Java) 700 行	(C 言語) 16 行
計算部分	570 行	400 行

8. まとめ

アプリケーション制作者が、容易に“アプリケーション処理の実行中に”3DCG オブジェクトの表示、操作あるいはパラメータの入力をマウスなどを用いて行う事が出来るための方法を提案した。本研究によれば、アプリケーション制作者が特別に3DCGの利用とGUI構築のためのプログラミング言語やその利用技術の学習を必要とせず、これらを用いたアプリケーションの製作が可能となることが期待される。

参考文献

- [1] 小林敏夫他：数値流体力学ハンドブック，丸善株式会社(2003)
- [2]<http://www.sw.nec.co.jp/library/jirei/jae/system.html> (PATRAS について)
- [3] エイリアスシステムス : Maya Complete 7.0.1 NL (Inventory Model)
- [4] デイ・ストーム : LightWave 3D
- [5] イーフロンティア : Shade 8.5 standard for Windows
- [6]<http://www.sw.nec.co.jp/APSOFT/SX/rvs/lib/> (RVSLIB について)
- [7] http://www.kgt.co.jp/avs_conso/ (AVS)
- [8] 戸松 豊和監訳 : Java ネットワークプログラミング第 2 版一，オライリー・ジャパン(2001)
- [9] 桑原邦郎，河村哲也 : 流体計算と差分法，朝倉書店(2005)

- [10] 篠田陽一 : 第 2 版 UNIX ネットワークプログラミング Vol.1, Vol.2, ピアソン・エデュケーション(2004)
- [11] 岩本信一訳 : 実践マルチスレッドプログラミング，アスキー出版局(1998)
- [12] 松田大樹 : Java によるジョブの多重実行支援環境，福井大学大学院工学部情報・メディア工学専攻博士前期課程学位論文(2006)
- [13] 犬塚義則 : グラフィックスライブラリの構成に関する研究，福井大学大学院工学部情報工学専攻修士課程学位論文(2002)
- [14] 俵谷和徳 : Java3D ワールド構築システムの研究，福井大学大学院工学部情報工学専攻修士課程学位論文(2002)
- [15] 日本バーチャルリアリティ学会第 6 回講演論文集(2001)
- [16] 日本バーチャルリアリティ学会第 8 回講演論文集(2003)
- [17] 日本数値流体力学会有限要素法研究委員会編 : 有限要素法による流れのシミュレーション，シュプリンガー・フェアラーク東京(2002)
- [18] 武宮博，田中良夫，中田秀基，関口智嗣 : Ninf-G2:大規模 Grid 環境での利用に即した高機能，高性能 GridPRC システムの実装と評価，情報処理学会論文誌，Vol.45, SIG 11(ACS 7),144-159(2004)
- [19] 住元真司，成瀬彰，九門耕一，細江広治，清水俊幸 : PM/InfiniBand-FJ:InfiniBand を用いた大規模 PC クラスタ向け高性能通信機構の設計，情報処理学会論文誌，Vol.45, SIG 11(ACS 7), 445-454(2004)
- [20] Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman : Compilers Principles, Techniques, and Tools, ADDISON-WESLEY PUBLISHING COMPANY(1986)
- [21] Edited by L. Rosenblum : SCIENTIFICVISUALIZATION Advances and Challenges, ACADEMIC PRESS(1994)