

4次元シミュレーションデータビューアの開発及び

仮想空間用 GUI の研究

Development of 4D Data Viewer - Graphical User Interface Library for Virtual Reality

杉山 大祐† 川原 慎太郎‡ 大野 暢亮‡ 坂下 善彦†

Daisuke Sugiyama Shintaro Kawahara Nobuaki Ohno Yoshihiko Sakashita

1. はじめに

現在、地球シミュレータに代表される大規模シミュレーションの可視化結果を分析するため、様々な VR 可視化システムが存在する。出力サイズが数 TB や数十 TB におよぶような大規模データの可視化結果は、高度に複雑な形状になる。このような形状を分析する際、VR 可視化システムが利用される。両眼立体視で捉えることにより、より直観的かつ効果的に現象の理解を促すことができるためである。

しかし、既存の VR 可視化システムは、VR 空間内でのグラフィカルユーザインターフェース (以下、GUI) 機能を持っていないものがほとんどである。表示したいファイルを切り替えたいとき、VR 空間用の GUI が無い場合は、一度 VR を終了し、VR 装置から出てコンソールから再度起動する必要がある。これは、大規模なデータサイズの結果をパラメータ設定しながら VR で閲覧・分析するような場合、非常に煩雑な作業である。これでは高い現実感を実現できる CAVE[1]のような没入型 VR 装置の長所である、仮想空間との対話性を十分生かしているとは言いがたい。パラメータ設定やファイルの切り替えなどを VR 空間で対話的に行うためには、VR 空間用の GUI を用意する必要がある。

しかしながら VR 空間用 GUI 機能がない VR ソフトウェアが多いのは、VR 空間用の GUI 機能がまだ一般化されていないためと考えられる。通常の平面ディスプレイ用であれば、3D の GUI も存在するが、あくまでマウスやキーなどの入力を想定しており、VR 空間用ではない。よって現状では、VR 空間用 GUI 機能を搭載するとき、VR ソフトウェア開発者は一から GUI を考え、GUI パーツとユーザが指し示すビームポインタとの衝突判定や、テキスト読み込み機能、空間への文字表示など、すべてをスクラッチから実装しなければならない。よって、本来目的とする機能の開発に注力できないために、VR 空間用 GUI 機能を搭載しないことを選択し、インタラクティブ性を低下させるという事態に繋がる。

VR 空間における GUI 機能をより容易に利用するためには、空間上で使いやすい GUI の研究と、ソフトウェアへ容易に空間 GUI を搭載することができる空間 GUI ライブラリの開発が必要不可欠である。

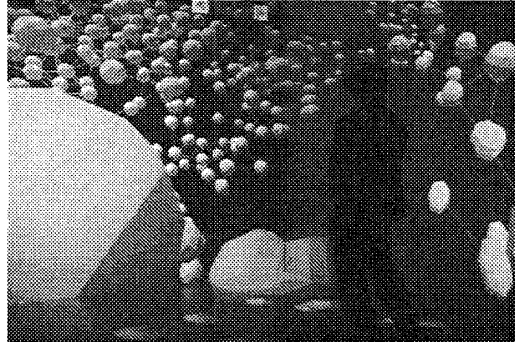


図1 4次元シミュレーションデータビューア

2006 年度に開発した 4 次元シミュレーションデータビューア(図 1)[2]には、表示したいファイルを VR 空間で選ぶ事ができる VR ファイルメニュー機能や、ボタン機能などの空間用 GUI 機能を実装した。この 4 次元シミュレーションデータビューアおよび VR ファイルメニュー機能は、新規に自作した Performer[3]ラッパライブラリ「pfclib」を使って実装している。しかし、現在においての CAVE 用 VR ソフトウェア開発の主流としてグラフィックスライブラリに OpenGL を使ったものと、その上位ライブラリ Performer を使ったものの 2 種類があり、これらはそれぞれ利用方法がまったく違うグラフィックスライブラリである。開発したラッパライブラリ「pfclib」の GUI 機能は Performer 用に作ったものであったため、OpenGL では使えないという問題があり、また、GUI 機能自体にも、より直感的にするべきなどの課題があった。

2. CAVEUI の開発

そこで本研究では、CAVE システムにおいて OpenGL と Performer、どちらの開発環境であっても同じように利用できる空間 GUI ライブラリ「CAVEUI」を新たに開発する。これを利用することで、CAVE システムを採用している VR 装置において、空間 GUI 機能の搭載を容易にすることを目的とする。

また、CAVEUI 実装例として、4 次元シミュレーションデータビューアを改良し、GUI 機能に CAVEUI を導入する。さらに 4 次元シミュレーションデータビューアの CAVEUI 導入後の GUI 機能と、CAVEUI 導入前の GUI 機能との使用感を比較することで、使用感の違いを検討する。

CAVEUI システム構成概要を図に示す。(図 2)

† 湘南工科大学大学院

‡ 地球シミュレータセンター

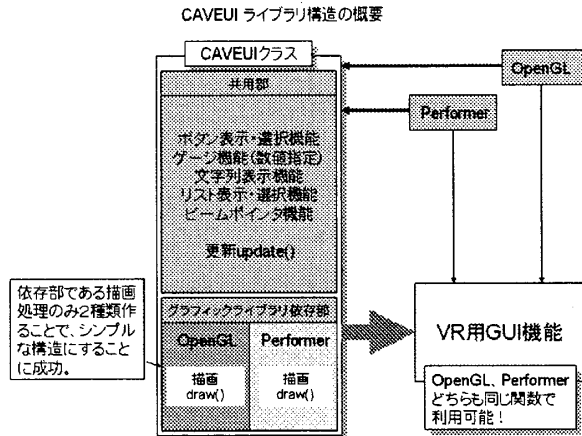


図2 CAVEUIライブラリのシステム構成

CAVEUIの開発にはC++言語を用い、CAVE表示用ライブラリとしてCAVELib[4]、グラフィックスライブラリとしてOpenGLと、Performerに対応した。

前述のように、OpenGLとPerformerはそれぞれ利用法が違うグラフィックスライブラリである。両者でGUIライブラリを実装するためには、2種類のライブラリを用意し、別々の利用法で使う必要があるが、これでは容易に利用できるとは言い難い。そこで、利用方法の差異をなくすために、GUIライブラリの機能実装方法について検討を重ね、GUIライブラリの機能を構成する部分のうち、グラフィックスライブラリに依存しなければならない部分は、リソースの確保と描画のみであることに着目した。この依存部のみ2種類作り、クラス内部でのみその依存部を呼び出すことで、両方の環境でほぼ差異のない利用方法を実現した。

また、実験的な機能として、ダイヤルを回すように手首の回転でメニュー項目を選択する、ダイヤルメニュー機能によって、ボタン式のメニュー選択機能より高速にメニュー項目を選択できることを目指した。

3. CAVEの概要

立体映像の提示に使用したCAVE型VR装置『BRAVE』*注1(図3)は、壁3面、床1面の4つのスクリーンを有し、それぞれのスクリーンにはプロジェクタにより視差画像が投影される。今回使用した装置は、スクリーンのサイズが一辺3mと大型のものであり、CAVE内に構築された仮想空間に対し、高い没入感を得ることができる。ユーザはCAVE内で液晶シャッターメガネを装着し、これを通してスクリーンに投影された映像を観察することで立体視を行う。この際、画像処理を用いたトラッキングにより、液晶シャッターメガネの位置および方向がリアルタイムで検出されるため、ユーザに対して常に正しい視差画像が提示される。視差画像の作成には、Onyx3800を用いた。

仮想空間とのインタラクションは、専用のコントローラを用いて行う。このコントローラについても液晶シャッターメガネと同様の位置・方向のトラッキングが行われる。

*注1『BRAVE』：地球シミュレータセンター提供

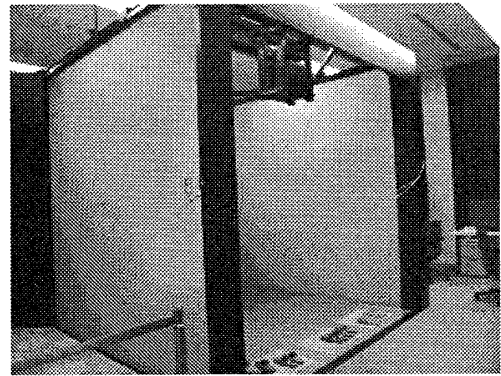


図3 CAVE型VR装置

4. 表示例

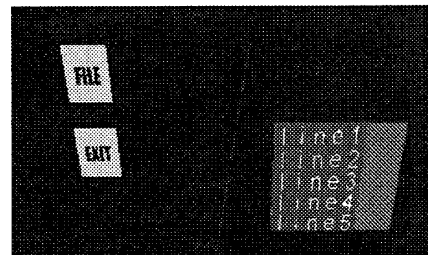


図4 CAVEUI表示例1

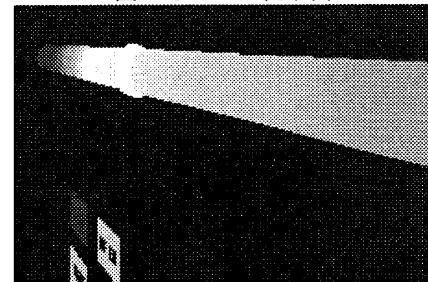


図5 CAVEUI表示例2

5. まとめ

CAVEUIを使い、4次元シミュレーションビューアに容易に仮想空間GUIを導入することができた。本研究ではCAVEシステムを使用しているが、今後はより広いデバイスの仮想空間GUIの研究を視野に入れ、CAVEUIの開発を通じて空間上で万人が直感的に解り易いGUI機能の研究を行っていく。

参考文献

[1] Cruz-Neira, C. Sandin, D. J. and DeFanti, T. A. "Surrounded-Screen Projection-Based Virtual Reality, The Design and Implementation of the CAVE", ACM SIGGRAPH 93, pp.135-142. 1993.
 [2] 4次元シミュレーションデータビューアの開発 杉山大祐, 陰山聡, 川原慎太郎, 坂下善彦, 2006
 [3] Silicon Graphics. OpenGL Performer Getting Started Guide. 1997, 2000, 2002-2004.
 [4] 陰山聡, 佐藤哲也. VRシステム CompleXcope プログラミングガイド, Research Report NIFS-MEMO No.28. 1998.