

VR Jugglerを用いたPCクラスタによる没入型ディスプレイの構築

浅井 紀久夫¹⁾

Construction of Immersive Display by PC Cluster with VR Juggler

Kikuo ASAI

要 旨

PCクラスタを構成することにより、没入型VR環境を構築した。システムは、低価格、アプリケーション構築の容易性、システムのスケーラビリティといった設計指針に基づいて実装された。GPU搭載のPCを4台つなげて、立体視映像を同期生成する。没入型VR環境を構築するためのフレームワークとして、VR Jugglerを利用した。複数のPCをネットワーク接続し、同期的にグラフィックス・レンダリングを行う機能を提供する。仮想世界のシーンを構築するフレームワークとして、OpenSceneGraphを利用した。教育応用を目的として、ラムサール条約に登録されている干潟を自由に動き回る体験ができる可視化アプリケーションを開発した。

ABSTRACT

An immersive VR (virtual reality) environment system was reconstructed by forming a PC cluster. The system was implemented on the basis of design guidelines including low cost, ease of making applications, and scalability for systems. Four PCs equipped with GPUs process the graphics rendering of stereoscopic images synchronously. The VR Juggler provides the synchronous graphics rendering with multiple PCs connected through a network, creating immersive VR environments. The OpenSceneGraph was used as a framework for creating 3D scenes of virtual environments. We developed a visualization application for educational purposes, enabling us to virtually experience a walkthrough on the tideland registered as the Ramsar Convention on Wetlands.

1. はじめに

1990年代初頭、没入型投影ディスプレイがCAVEシステム^[1]として導入されてから、様々な用途の可視化環境を実現するために多くのシステムが構築された。典型的なCAVEシステムは、正面と側面の3つの壁面、それに1つの床面を持つ立方体スクリーンからなり、各スクリーンには立体視映像が投影される。従来のシステムでは、没入型VR (Virtual Reality) 環境を構成する立体視映像の生成に、専用のグラフィックス・ワークステーションを使ってきた。しかし、PCの性能向上に伴い、GPU (Graphics Processing Units) を搭載した複数のPCを使って負荷分散を行うことにより、グラフィックス・クオリティとレンダリング・スピードの改善が図られた^[2, 3]。最近のシステムは、初期のシステムより低価格で構築できるように

なり、アプリケーションや利用者が様々な領域に拡大してきた。低価格化の主な要因は、コンピュータ、周辺機器、ソフトウェア等が高価なものからコモディティ製品へと移行したことである。しかし、低価格で没入型VR環境を構築することには、未だ多くの労力と高いスキルが要求される。また、コンピュータ、グラフィックス・デバイス、プロジェクタ、スクリーンなどのハードウェアはただではない。本論文は関連研究を調査し、没入型VR環境を構築する際の要求レベルを軽減する工夫について整理する。

我々は、月面探査の可視化、世界遺産の探索、コンピュータ・プログラミングの学習といった教育支援のための没入型VR環境を構築してきた^[4]。このときのグラフィックス・システムは、SGIのグラフィックス・ワークステーションとOpenGL Performerベースの独自ライブラリで構成された。しかし、このシステムは、システム保守の高額コストとアプリケーション

¹⁾ 放送大学教養学部 (「情報」コース)



(a) 放送大学の没入型ディスプレイ



(b) 谷津干潟アプリケーション

図1 没入型 VR 環境 TEELeX

を作るための高いスキルを要求した。しかも、仮想環境を構築するための独自API (Application Programming Interface) は、異なるプラットフォームでは動作しなかった。その主な理由は、OSに対する柔軟性が欠けていることと、ステレオ・プロジェクタや入力デバイスのようなハードウェアに対するスケーラビリティが欠如していることであった。

そこで、今回、高価な保守や高度なスキルを必要としないように、没入型VR環境を再構築した^[5] (図1 (a))。システムは、低価格、アプリケーション構築の容易性、ハードウェアのスケーラビリティを含む設計指針に基づいて実装された。GPUを搭載した4台のPCでPCクラスターが構成され、ステレオ映像を同期して生成する処理を行う。没入型VR環境を作るためのフレームワークとして、VR Juggler^[6]を採用した。これはグラフィックス・レンダリングを十分なパフォーマンスで実行し、安定的に動作したからである。また、仮想世界のシーンを構築するフレームワークとして、OpenSceneGraph^[7]を利用した。そして、教育目的の可視化アプリケーションとして、ラムサール条約^[8]に登録されている谷津干潟を再現し、仮想的に歩き回ることができるようにした (図1 (b))。

2. 先行研究

没入型VR環境を構築するためのコストの制約は、様々な技術による多くの挑戦を生み出してきた。関連研究について、ディスプレイとソフトウェア構成の2つに分けて述べる。

2.1 ディスプレイ

最初のCAVEは没入型投影ディスプレイであり、大きなスクリーンによってユーザの視野を広く覆うことに特徴付けられる。専用の部屋が必要になるものの、ユーザはプロジェクタで投影された立体視映像によって仮想世界を広視野で見ることができる。大きなスク

リーンでユーザを取り囲むのは、臨場感を増加させることに加えて、画像生成の処理負荷を軽減することにメリットがあった。ユーザ頭部の並進運動は、新たな画像生成・描画を必要としない。そのため、CAVEシステムの多くは5面、6面というように立方体型スクリーンの面数を増加させる方向に発展してきた^[9, 10, 11, 12]。臨場性の創出は、ディスプレイが立体視映像でユーザの視野を覆うことによる。視覚的な臨場性は、仮想世界での作業や課題の効率を左右することがある^[13]。科学可視化のようなアプリケーションでは、ユーザが没入型ディスプレイの中を動き回りながら視覚的データを観察することにより、時に科学的発見に至ることもある^[14]。StarCAVEは5つの壁と1つの床にスクリーンを配置した没入型ディスプレイであり、視覚的な明るさと解像度において高い性能を達成している^[15]。

CAVEシステムを構築する際の最大の問題は価格である。そのため、低価格CAVEあるいはその派生システムを設計することに、多大な努力が重ねられてきた^[16, 17, 18, 19]。CAVEシステムを低価格で構築する要求は、没入型投影ディスプレイのスクリーン面の数を減らして小型化するという方向性を生み出した。その典型例は、一連の単一面ディスプレイである。これらは単一面ではあるものの、後方投影による立体視映像を提示する^[20, 21]。これらのディスプレイは視野が狭いものに制限されるが、この制限が利用性に対して常に障壁になるとは限らない。利用性への影響は、どのような利用をするかによる^[13]。

投影ディスプレイにおいて1面と4面あるいはそれ以上の面数との折衷案として、2面ディスプレイが提案された^[22, 23]。2面ディスプレイは視野角の制限を緩めることができるが、価格とシステムの複雑さが増すのは避けがたい。典型的な2面ディスプレイでは、2つのスクリーンが互いに直角に配置される。そうした配置は、2つのスクリーンの前方にプロジェクタを置くことを空間的に容易にする。前方投影は、普通のス

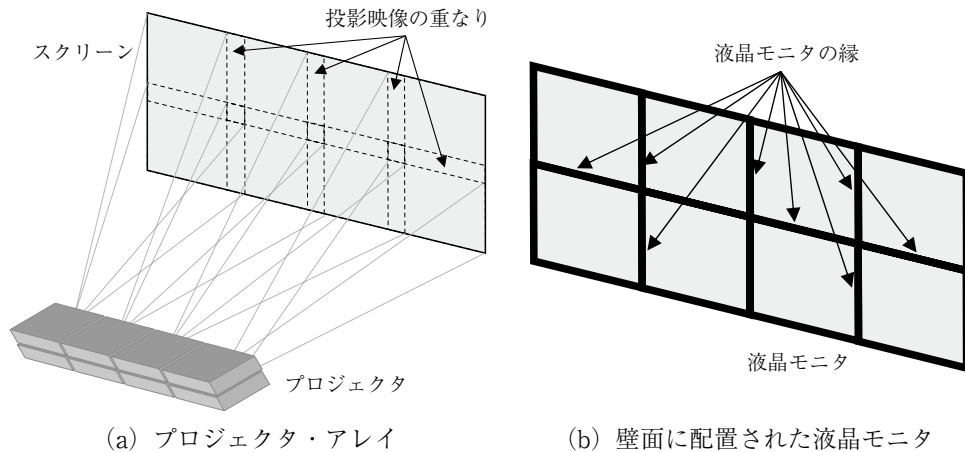


図 2 Tiled ディスプレイ

クリーンの代わりに部屋の角面を投影面に使うことによって後方投影よりも利便性が高く、省スペースであり、部屋の大きさの条件を大きく緩和する。しかしながら、前方投影システムは、前方投影された映像がユーザによって遮蔽される可能性があり、ディスプレイ内でのユーザの動きの自由度が制限される。LAIRシステム^[24]は四角い部屋の4つの壁と8つのプロジェクタを使って映像を前方投影し、立体視には液晶シャッター眼鏡で対応している。2つのプロジェクタを壁面間の中央からズラした位置に配置することによって、前方投影の欠点を最小化している。2つのプロジェクタはそれぞれ、壁の領域半分を投影する。そのため、部屋の中央付近に立つユーザが、投影映像を遮蔽することはない。

Tiledディスプレイは、解像度、輝度、コストといった、CAVEシステムが持つ諸問題への解決策を提供する^[25]。Tiledディスプレイはユーザの視野を完全には覆わないが、大きな提示領域を提供する。大型ディスプレイを使ったユーザ行動に関する多くの調査は、大きな提示領域が視覚的情報の探索、空間的方向感覚、日常業務のような作業におけるユーザ・パフォーマンスを改善するという仮説を支持してきた^[26, 27, 28]。Tiledディスプレイは、基本的に2つのアプローチに分類される。プロジェクタ・アレイ (図 2 (a)) と壁面に配置された液晶モニタ (図 2 (b)) である。

プロジェクタ・アレイは複数のプロジェクタを並列に配置することにより構成され、慎重な校正によってスクリーン間に完全につなぎ目のない映像を提供する^[29, 30]。プロジェクタ・アレイは投影のための大きな物理的空間を必要とし、色、輝度、位置合わせといった投影映像間のエッジ・ブレンディングの手間を要する^[31, 32, 33]が、つなぎ目のない映像は視覚的な臨場感にとって重要である。

壁面配置の液晶モニタは多数の液晶モニタを配置することにより構成され、普及製品を活用することによる高解像度及び高輝度の映像、低価格を提供する。ただ、各液晶モニタの縁による映像の不連続が生じると

いう欠点がある。先行の評価実験は、モニタの縁がユーザの探索戦略やマウス操作に影響することが報告されている^[34]。そのため、プロジェクタと液晶モニタの組み合わせが縁問題を解決する方法として提案されている^[35]。

2.2 ソフトウェア構成

CAVEシステムが提案された当初は、複数のグラフィックス・エンジンを搭載した高価なワークステーションによって3DCGのレンダリングを行っていた。アプリケーションを開発するには、プログラミングの高いスキルと考察が要求され、計算機システム、グラフィックス・エンジン、インタフェース・デバイスのようなハードウェアに強く依存したプログラム構成にする必要があった。低価格化は、グラフィックス・ワークステーションの代わりにPCをネットワークに接続し、3DCGレンダリングの負荷を分散する並列化処理の実現と共にもたらされた^[36]。最近のPCは高性能なGPUを搭載しており、1台でもデスクトップ・コンピュータ環境での複数モニタのための高解像度画像を描画することができる。しかも、複数モニタを構成するために、複雑な設定は必要ない。

ここでは、ネットワーク接続された一連のコンピュータで構成されるPCクラスタを使って、3DCGの並列レンダリングを行うための仕組みを概観する。この仕組みは主に、2つのタイプに分類される。一つは画像ベース・ストリーミングで、もう一つは分散レンダリングである。

画像ベース・ストリーミングは、2次元画像やビデオのような画像データを遠隔ディスプレイに配信することが特徴である。画像ベース・ストリーミングは、イリノイ大学シカゴ校電子可視化研究所によって行われた一連の研究が基になっている。彼らは、TeraVision^[37]、SAGE^[38]、JuxtaView^[39]、TeraScope^[40]といった協調可視化環境を構築しようとして、様々な可視化アプリケーションによって生成された画像を遠隔地の間で共有するという画像ベース・ストリーミン

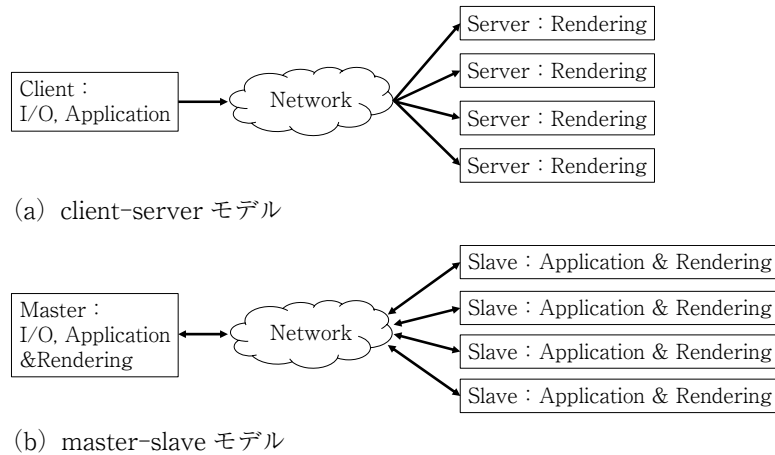


図3 分散レンダリングのモデル

グの基本的概念を確立した。協調可視化は、高速ネットワークを通してクラスタ・ノードに画像を配信し、複数のアプリケーションに対してTiledディスプレイ上に独立に画像を表示することにより達成される。各クラスタ・ノードは、同期メッセージで同期されている。SAGEは多くのクラスタ・ノードを要求し、その数はTiledディスプレイの液晶モニタの数に比例する。

分散レンダリングは、各クラスタ・ノードで画像をレンダリングすることにより特徴づけられる。レンダリングはアプリケーションのデータをクラスタ・ノードに配信し、シーングラフ、グラフィックス・コマンド、描画領域、同期信号といったレンダリング情報を提供することにより実行される^[41]。クラスタ・コンピューティングにおける分散レンダリングは、データ配信という観点で研究され、その配信方法が描画性能に影響を与えると予想された^[42, 43]。分散レンダリングは、データ配信の方法によって2つの実行モデルに分類される。一つはclient-serverモデル（図3 (a)）であり、もう一つはmaster-slaveモデル（図3 (b)）である。

client-serverモデルは単一のクライアント・ノードでアプリケーションを実行し、幾何学データのようなレンダリング情報を生成する。クライアントで生成されたレンダリング情報は、複数のレンダリング・サーバに転送される。このレンダリング・サーバは、ネットワーク上にあって各サーバ・ノードで同期的に画像をレンダリングする。client-serverモデルは分散レンダリングに伴うアプリケーションのソース・コードの修正を要求しないが、実行速度の観点でネットワークにボトルネックを生じる可能性がある。例えば、複雑なシーンでレンダリング情報が急速に増えた場合、ネットワークに過剰な負荷がかかる。代表的なソフトウェアにはChromium^[44]やOpenSG^[45]があり、前者はサーバ・ノードに配信されるレンダリング情報としてOpenGLコマンドを、後者はシーングラフの変更リストを使っている。

master-slaveモデルは全スレーブ・ノード上でアプリケーションを実行し、そのノード間で同期して画像

をレンダリングする。マスタ・ノードは画像をレンダリングする処理に先立って、スレーブ・ノードにアプリケーション・データを配信する。描画画像の一貫性は、スレーブ・ノード間でアプリケーションの実行状態が同期することを要求する。同期は、アプリケーションの状態を変更するユーザ入力やタイマーといったデータの転送によって実行される。master-slaveモデルはネットワーク負荷に伴う実行速度のボトルネックを生じにくい、分散レンダリングのためのアプリケーションのソース・コードの修正を要求する。代表的なソフトウェアとして、CAVELib^[46]やVR Juggler^[47]、Syzygy^[48]がある。

master-slaveモデルの問題の一つは、アプリケーションのソース・コードを修正するためのプログラミング作業が増加することである。そのアプリケーションには、並列レンダリングのための同期処理が、ライブラリやアプリケーション・フレームワークを使って実装される。API割り込み技術が、スレーブ・ノード間でアプリケーションの実行状態の同期を保持するための代替策として提案された^[49]。この方法では、同期の仕組みがアプリケーションの代わりにAPIに移行され、アプリケーション側でソース・コードを修正しなくても済むようにしている。

3DCGの並列レンダリングにおけるクラスタ・コンピューティングのもう一つの問題は、クラスタ・ノード間のレンダリング負荷の不均衡である。負荷の不均衡はしばしば、ユーザの視点に対して正面方向のシーンが非常に複雑で、他の方向のシーンにはほとんど何も存在しないという場合に起こる。これは、それぞれのシーンが別々のクラスタ・ノードでレンダリングされるためである。コンピュータ・クラスタでの負荷均衡の仕組みが提案され、各クラスタ・ノードへの負荷を動的に割り当てることによって並列レンダリングの性能が改善された^[50, 51]。

上記に加えて、ゲーム・エンジンは、没入型VR環境を構築するためのプログラミング・ツールとして便利な機能を提供する。ほとんどのツールはユーザ・イ

インタラクションや複数ディスプレイを設計するための編集機能を持っており、アプリケーション構築に便利である。そのようなゲーム・エンジンには、Unreal Tournament^[52]やCryEngine^[53]、HalfLife^[54]がある。

3. システム設計

我々の没入型VR環境はTEELeXと名付けられ、元々CAVE型システムとして開発された^[4]。5.5面立方体スクリーンを持つ投影ディスプレイであり、立体視映像を提示した。このシステムを再構成するに際し、低価格と容易なメンテナンス、アプリケーション構築の容易性、システム構成のスケーラビリティといった設計指針に沿うようにシステム・インテグレーションを行った。ここでは、システム設計について投影ディスプレイ、システム構成、ソフトウェア構成の観点から記述する。

3.1 投影ディスプレイ

図4に、没入型VR環境の投影ディスプレイの構成を示す。透過型スクリーンと液晶プロジェクタは、仮想環境の立体視映像を提示するために再利用された。立方体スクリーンの5.5面のうち4面が、投影ディスプレイとして使用された。4面だけを利用したのは、

プロジェクタのランプを減らしてコストを削減するためである。そのため、各スクリーンに対して立体視映像を作り出すには、8つのプロジェクタを用意すれば済む。立体視の左目と右目の映像は円偏光を通して互いに区別される。各面の正方形スクリーンは大きさ3m×3mであり、プロジェクタは解像度1,000×1,000ピクセルで映像を投影する。各液晶プロジェクタは、輝度1,000 ANSI-lmを持つ。

3.2 システム構成

図5に、没入型VR環境のシステム構成を示す。4つのスクリーンへの立体視映像の投影は、同時に8つの画像を生成する必要がある。クラスタ・コンピューティングにより、3DCGの並列レンダリングを実現した。コストと保守を考慮し、1つのPCをマネージャ（管理PC）に割り当て、GPUを搭載した4つのPCをそれぞれレンダリング・スレーブマシン（レンダリングPC）に割り当てた。管理PCは、レンダリングPC及びアプリケーション・データを管理する。各レンダリングPCは立体視のための左目と右目の画像を同期生成し、GPUからビデオ信号を出力する。スイッチングHUBで、ギガビット・ネットワークを通して管理PCとレンダリングPCが接続される。

ユーザ・インタラクションを支援するため、インタ

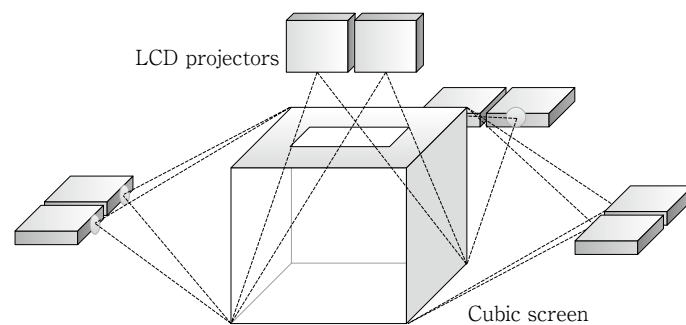


図4 没入型投影ディスプレイの構成

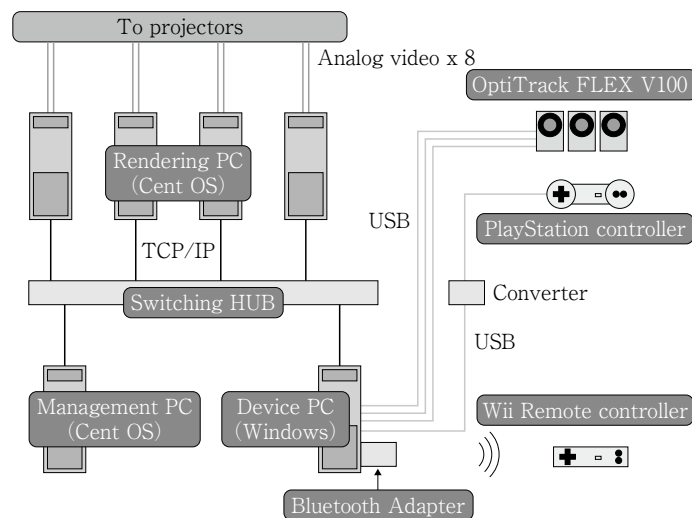


図5 システム構成

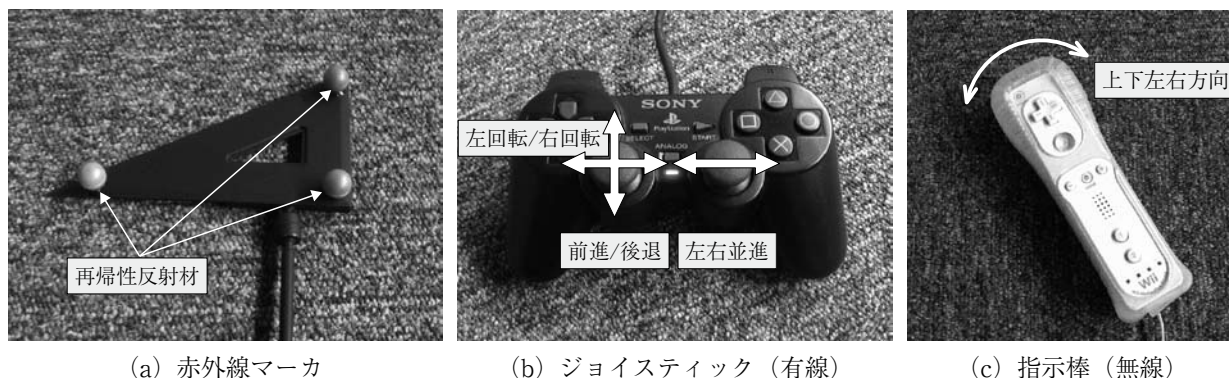


図6 ユーザ・インタフェース

フェース・デバイスが3つ（頭部位置姿勢センサ、ジョイスティック、指示棒）、没入型VR環境のシステムに組み込まれた。ユーザの頭部運動は、視点に基づいて合成画像を生成するために必要である。そこで、ユーザの頭部運動を赤外線センサ（OptiTrack^[55]）によって追跡することにより、ユーザは自らが好む視点から仮想世界を見ることができる。三角測量の原理により、3つの球からなる剛体の幾何学に基づいてユーザ頭部の位置と姿勢が推定される。赤外線センサで検出できるようにするため、3つの球は図6 (a) に示すように再帰性反射材で覆われている。図6 (b) に示したような有線ゲーム・パッド（PlayStation2）がジョイスティックとして、3次元空間内における視点を制御するために使われる。加速度計を搭載した無線ゲーム・コントローラ（Wii Remote）（図6 (c)）も、仮想物体を操作するための指示棒として使われた。こうしたインタフェース・デバイスはデバイス制御用PCによって制御され、インタラクション・データはギガビット・ネットワークを通して管理PCに送られる。

3.3 ソフトウェア構成

図7に、没入型VR環境を構築するためのソフトウェア構成を示す。VR Juggler^[6]が、クラスタ・コンピューティングを使った並列レンダリングのためのフレームワークとして採用された。また、VRPN^[56, 57]が、ネットワークを介して様々なインタフェース・デバイスをVRアプリケーションに実装するためのライブラリ及びサーバとして使われた。CppDom^[58]は、VR Jugglerで使われるXMLパーサである。GMTL^[59]はGraphics Math Template Libraryを表し、グラフィックスと数学演算の基本的なクラス群を提供する。これも、VR Jugglerにおいて使われる。Boost^[60]はC++のオープン・ソース・ライブラリで、アルゴリズムやコンテナ、並列プログラミングに関する様々な標準機能を提供する。Linuxオペレーティング・システムとして、CentOS 6^[61]が採用された。

アプリケーションとコンテンツのデータは、管理PCに蓄積される。各レンダリングPCは、NFS

（Network File System）を通してこれらのデータを参照する（図8）。VR Jugglerには管理PCからレンダリングPCへのデータ転送を行う枠組みがあるが、NFSはユーザがデータ転送について考慮しなくて済むようにできる。

シーングラフに基づいた仮想環境を記述するために、OpenGLで動作するオブジェクト指向フレームワークとしてOpenSceneGraph^[7]が採用された。可視化アプリケーションは、OpenSceneGraphに基づいて実装される。そのため、谷津干潟アプリケーションやOSG Viewer（シーングラフを表示するビューワ）は、VR JugglerとOpenSceneGraphに依存することになる。OpenSceneGraphを使うことは、アプリケーションが異なるプラットフォームで動作するのを容易にし、パラメータ設定で制御できるようになる。全くスキルを要求しないというわけにはいかないが、要求レベルを大幅に下げることができる。

4. 実装

クラスタ・コンピューティングのクラスタ・ノードとして、PCを使った。CPUにIntel Core i7-2600 3.4 GHzのプロセッサを、メインメモリに4 GBのDDR3 SDRAMを、GPUにnVidia GeForce GTX 560 Tiを搭載したPCを用意し、3DCGの並列レンダリングのプラットフォームとした。表1に、クラスタPC及び周辺機器の仕様を示す。

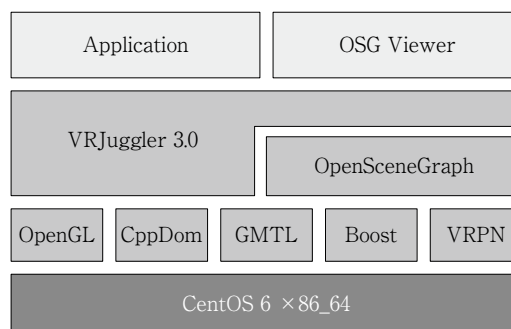


図7 ソフトウェア構成

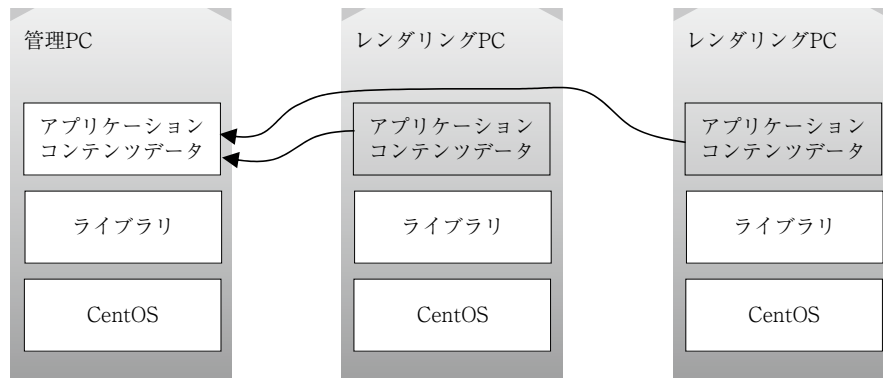


図8 管理PCとレンダリングPC

表1 PC及び周辺機器の仕様と性能

Equipment	Specifications
CPU	Intel Core i7-2600 3.4 GHz
Memory	4 GB DDR3 SDRAM
GPU	nVidia GeForce GTX 560 Ti
OS	Linux CentOS 6
Network	Gigabit Ethernet LAN
I/O device	OptiTrack FLEX : V100, PlayStation2 controller, Wii Remote controller
Equipment	Performance
Frame rate	146.2 f/s (Average of 17 steps)
CPU	72.5/800 % (Management PC), 83.1/800 % (Rendering PC)
Memory	427 MB (Management PC), 649 MB (Rendering PC)
Network	179 KB/s (Receive at Management PC), 1.5 MB/s (Send at Management PC)

VR Jugglerは、作業開始のマニュアルによれば、以下の手順でインストールされる。1) shell scriptを実行するためのmakefileの作成、2) buildの作成、3) そのインストールである。実際には、build環境を構築するのに多くの時間がかかった。これは、VR Jugglerのbuildにおいて依存関係にあるライブラリが多数存在するためである。依存ライブラリのインストール手順を、付録に記述した。VR Jugglerのインストール後、プラットフォームに適したパラメータをセットするためにconfiguration fileを作る必要がある。これは、VR Jugglerが我々の没入型VR環境のシステムでうまく動作するように設定しておくためのものである。この作業はVR Jugglerの設計上の柔軟性とスケーラビリティの結果ではあるものの、configuration fileで多数のパラメータを設定することは簡単ではない^[62]。configuration fileでのパラメータは主に、ディスプレイ・システム、ディスプレイ・ウィンドウ、面表示域（ビューポート）から構成される。表2に、configuration fileの例を示す。付録の表Aに、configuration fileにおける代表的パラメータの設定例を列挙する。

アプリケーションを実行するモードとして、以下の組み合わせを用意した。管理PC及びレンダリングPCで実行されるコマンドとそのオプションを示す。ここでは、osgTEELeXが実行ファイルにあたり、main.osg

がシーングラフを記述したファイルである。図9に、“TEELeXの4面表示と頭部追跡+ジョイスティック”モード時の動作の様子を示す。管理PCの表示画面（図9（a））にあるレンダリング・モニタは、レンダリングPC（図9（b））で描画されるべき画像を示している。

TEELeXの4面表示と頭部追跡+ジョイスティック

実際に、TEELeXの4面スクリーンに立体視映像を提示する。ユーザの頭部を追跡して、その視点に整合した映像を生成する。また、入力デバイスとしてジョイスティックを使用する。

・管理PC

```
./osgTEELeX [フォルダ名]/main.osg --vrjmaster
vrpnHead.jconf teejex4.jconf
```

・レンダリングPC

```
./osgTEELeX [フォルダ名]/main.osg --vrjslave
--listen_port 7000
```

TEELeXの4面表示とデバイス・シミュレータ

実際に、TEELeXの4面スクリーンに立体視映像を提示する。入力デバイスの実機は使わず、アプリケーションを実行し、操作はキーボードで行う。

・管理PC

```
./osgTEELeX [フォルダ名]/main.osg --vrjmaster
```

simHead.jconf teejex4.jconf

・レンダリングPC

./osgTEELeX [フォルダ名]/main.osg --vrjslave
--listen_port 7000

TEELeXの4面シミュレータとデバイス・シミュレータ

実際には、TEELeXの4面のスクリーンには映像を提示しない。管理PC内で4面映像の表示を確認する。また、入力デバイスの実機も使わず、アプリケーションを実行し、操作はキーボードで行う。

・管理PC

./osgTEELeX [フォルダ名]/main.osg simHead.jconf
simTeejex4.jconf

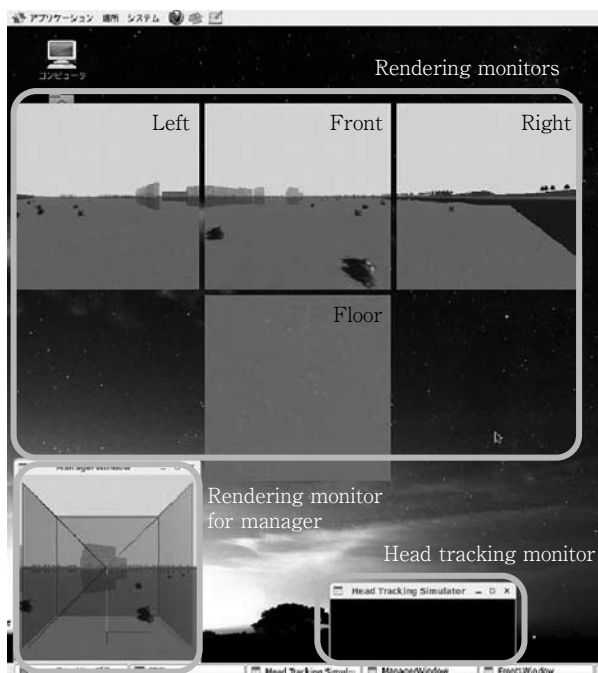
5. コンテンツの開発

没入型VR環境TEELeXを使ったインタラクティブ・コンテンツを構築するには、VR Juggler及びOpenSceneGraphのフレームワークに基づくアプリケーション・プログラムを作成する。本アプリケーションは、1) インタクションのための基本機能と2) 仮想空間内のウォークスルーといった視点制御機能を提供する。コンテンツ用の3Dシーングラフのデータとして、OpenSceneGraphがサポートする全てのファイル形式を利用できる。

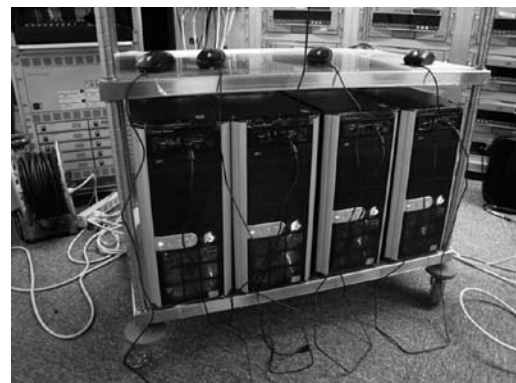
本アプリケーションに読み込まれるOpenSceneGraph

表2 configuration fileの例

Configuration file	構成
simHead.jconf	頭部追跡をシミュレートするための構成ファイル：デバイスを使用しないで、アプリケーションを実行する場合に利用する
simTeelex1.jconf	管理PC内でTEELeXの正面1面を表示するための構成ファイル：立体視映像は生成しない。TEELeXレンダリング・ウィンドウ1つと管理用ウィンドウが表示される。
simTeelex4.jconf	管理PC内で正面、左右面、床面スクリーンのTEELeX用4面を表示するための構成ファイル：立体視映像は生成しない。TEELeXレンダリング・ウィンドウ4つと管理用ウィンドウが表示される。
teelex1.jconf	TEELeXの正面スクリーンのみを使うための構成ファイル。立体視映像を生成する。管理用ウィンドウが表示される。
teelex4.jconf	TEELeXの正面、左右面、床面スクリーンを使うための構成ファイル。立体視映像を生成する。管理用ウィンドウが表示される。
vrpnHead.jconf	VRPN経由でデバイスPCからインタフェース・デバイス（頭部位置姿勢センサ、ジョイスティック）のデータを取得するための構成ファイル。
vrpnHead_wii.jconf	VRPN経由でデバイスPCからインタフェース・デバイス（頭部位置姿勢センサ、ジョイスティック、指示棒）のデータを取得するための構成ファイル：Wii Remoteを利用している場合は、この構成ファイルを利用する。



(a) 管理 PC の表示画面

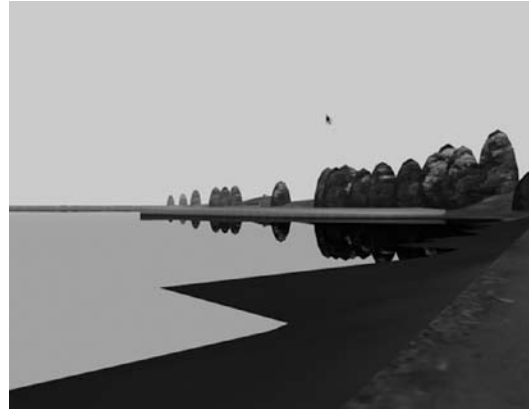


(b) レンダリング PC

図9 動作の様子



(a) 谷津干潟



(b) 合成画像

図 10 谷津干潟コンテンツ

ファイルの中にノードを追加することにより、アプリケーションに対する設定が可能である。設定項目として、背景色や霧、視点及び姿勢の初期状態、水面の定義が含まれる。また、仮想空間内のインタラクションは、拡張ノードを記述することにより定義する。拡張ノードには、注釈付与、音響出力、メニュー選択、外部モデル参照、シーン切替、タイマーといった機能が含まれる。こうした機能を適宜組み合わせることにより、様々なアプリケーションを構築することができる。

独自アプリケーションのデモとして、ラムサール条約（湿地の保存に関する国際条約）に登録されている谷津干潟を可視化するコンテンツを作成した。この可視化アプリケーションでは、ユーザが谷津干潟の景色を眺めたり、湿地帯を動き回ったりできるようになっており、仮想的な体験を提供する。図10に、(a) 谷津干潟の写真と (b) レンダリングPCの一つによって生成された合成画像を示す。また、このコンテンツでは、谷津干潟の現在と1970年代頃の様子を見比べたり、野鳥を観察したりできるようになっている。

6. まとめ

PCクラスタと無料で提供されているソフトウェアを使って没入型VR環境システムを構築した。本論文の中では、システム設計において複数ディスプレイと、クラスタ・コンピューティングによる並列3DCGレンダリングのためのソフトウェア構成についてサーベイした。今後、可視化アプリケーションの開発及びその評価を行っていく予定である。

7. 謝辞

本研究の一部は、放送大学学長裁量経費の支援によって実施された。

8. 参考文献

- [1] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, Surround-screen projection-based virtual reality : the design and implementation of the CAVE, Proc. ACM SIGGRAPH, pp.135-142, 1993.
- [2] T. Ni, G. S. Schmidt, O. G. Staadt, M. A. Livingston, R. Ball, and R. May, A survey of large high-resolution display technologies, techniques, and applications, Proc. IEEE Conference on Virtual Reality, pp.223-236, 2006.
- [3] L. P. Soares, B. Raffin, J. A. Jorge, PC clusters for virtual reality, International Journal of Virtual Reality vol.7, pp.67-80, 2008.
- [4] K. Asai, N. Osawa, and Y. Y. Sugimoto, Virtual environment system on distance education, Proc. EURO-MEDIA'99, pp.242-246, 1999.
- [5] K. Asai, and N. Takase, Reconstructing an immersive environment system with PC Cluster and freely available software, Proc. International Workshop on Advanced Image Technologies (IWAIT), 6 pages, 2013.
- [6] VRJuggler, <http://code.google.com/p/vrjuggler/>
- [7] OpenSceneGraph, <http://www.openscenegraph.org/>
- [8] Ramsar Convention, <http://www.ramsar.org/>
- [9] T. Yamada, M. Hirose, and Y. Iida, Development of a complete immersive display : COSMOS, Proc. International Conference on Virtual Systems and Multimedia, pp.522-527, 1998.
- [10] M. Hirose, T. Ogi, S. Ishiwata and T. Yamada, Development and evaluation of CABIN immersive multi-screen display, Systems and Computers in Japan, Scripta Technica, vol.30, pp.13-22, 1999.
- [11] J. Ihrén, and K. J. Frisch, The fully immersive CAVE, Proc. International Immersive Projection Technology Workshop, pp.59-63, 1999.
- [12] M. Robinson, J. Laurence, A. Hogue, J. Zacher, A. German, and M. Jenkin, IVY : Basic design and construction details, Proc. International Conference on Artificial Reality and Telexistence, pp.30-35, 2002.
- [13] D. A. Bowman and R. P. McMahan, Virtual Reality :

- How much immersion is enough?, IEEE Computer, vol.40, pp.36-43, 2007.
- [14] N. Ohno, and A. Kageyama, Region-of-interest visualization by CAVE VR system with automatic control of level-of-detail, Computer Physics Communications, vol.181, pp.720-725, 2010.
- [15] T. A. DeFanti, G. Dawe, D. J. Sandin, J. P. Schulze, P. Otto, J. Girado, F. Kuester, L. Smarr, and R. Rao. The StarCAVE, a third-generation CAVE and virtual reality OptiPortal, Future Generation Computer Systems, vol.25, pp.169-178, 2009.
- [16] W.-J. Lia, C.-C. Changb, K.-Y. Hsub, M.-D. Kuoc, D.-L. Wayd, A PC-based distributed multiple display virtual reality system, Displays, vol.22, pp.177-181, 2001.
- [17] A. Peternier, S. Cardin, F. Vexo, D. Thalmann, Practical design and implementation of a CAVE system - high quality four walls CAVE howto, Proc. International Conference on Computer Graphics Theory and Applications (GRAPP), pp.129-136, 2007.
- [18] S. Schlechtweg, miniCAVE-a fully immersive display system using consumer hardware, Proc. Eurographics Conference on Virtual Environments (EGVE), pp.73-80, 2008.
- [19] R. H. Fowler, M. D. Carrillo, R. A. Huerta, W. A. L. Fowler, Designing a low cost immersive environment system : twenty years after the first CAVE, Proc. International Conference on Computer Graphics and Virtual Reality, WORLDCOMP, 2012.
- [20] M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, G. L. Dawe, and M. D. Brown, The ImmersaDesk and Infinity Wall projection-based virtual reality displays, ACM SIGGRAPH Computer Graphics, vol.31, pp.46-49, 1997.
- [21] W. Krueger, and B. Froehlich, The Responsive Workbench, Journal of IEEE Computer Graphics and Applications, vol.14, pp.12-15, 1994.
- [22] H. J. Gardner, R. W. Boswell, and D. Whitehouse, The WEDGE emmersive projection theatre, Proc. International SimTecT Conference, pp.383-385, 1999.
- [23] P. M. Sauter, VR2GoTM : a new method for virtual reality development, ACM SIGGRAPH Computer Graphics, vol.37, pp.19-24, 2003.
- [24] B. Denby, A. G. Campbell, H. Carr, and G. M. P O'Hare, The LAIR : Lightweight Affordable Immersion Room, Presence : Teleoperators and Virtual Environments, vol.18, pp.409-411, 2009.
- [25] G. Wallace, O. J. Anshus, P. Bi, H. Chen, Y. Chen, D. Clark, P. Cook, A. Finkelstein, T. Funkhouser, A. Gupta, M. Hibbs, K. Li, Z. Liu, R. Samanta, R. Sukthankar, and O. Troyanskaya, Tools and applications for large-scale display walls, IEEE Computer Graphics and Applications, vol.25, pp.24-33, 2005.
- [26] D. S. Tan, D. Gergle, P. Scupelli, and R. Pausch, Physically large displays improve performance on spatial tasks, Transactions on Computer-Human Interaction, vol.13, pp.71-99, 2006.
- [27] R. Ball and C. North, Visual Analytics : Realizing embodied interaction for visual analytics through large displays, Computers and Graphics, vol.31, pp.380-400, 2007.
- [28] X. Bi and R. Balakrishnan, Comparing usage of a large high-resolution display to single or dual desktop displays for daily work, Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), pp.1005-1014, 2009.
- [29] C. Li, H. Lin, and J. Shi, A survey of multi-projector tiled display wall construction, Proc. International Conference on Image and Graphics, pp.452-455, 2004.
- [30] N. Hashimoto, J. Ryu, M. Yamasaki, T. Minakawa, H. Takeda, S. Hasegawa, M. SATO, D-vision : Immersive multi-projection display with a curved hybrid screen, Proc. International Conference on Virtual Systems and Multimedia (VSMM), pp.192-201, 2002.
- [31] M. C. Stone, Color and brightness appearance issues in tiled displays, IEEE Computer Graphics and Applications, vol.21, pp.58-66, 2001.
- [32] Y. Chen, D. W. Clark, A. Finkelstein, T. C. Housel, and K. Li, Automatic alignment of high-resolution multi-projector display using an un-calibrated camera, Proc. IEEE Conference on Visualization (VIS '00), pp.125-130, 2000.
- [33] E. S. Bhasker, P. Sinha, and A. Majumder, Asynchronous distributed calibration for scalable and reconfigurable multi-projector displays, IEEE Transactions on Visualization and Computer Graphics, vol.12, pp.1101-1108, 2006.
- [34] X. Bi, S.-H. Bae, and R. Balakrishnan, Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection, Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), pp.65-74, 2010.
- [35] A. Ebert, S. Thelen, P. Olech, J. Meyer, and H. Hagen, Tiled + + : an enhanced tiled hi-res display wall, IEEE Transactions on Visualization and Computer Graphics, vol.16, pp.120-132, 2010.
- [36] M. Song and S. Park, A survey on projector-based PC clustered distributed-rendering large screen displays and techniques, Proc. International Conference on Entertainment Computing, LNCS vol.4740, pp.445-449, 2007.
- [37] R. Singh, Byungil Jeong, L. Renambot, A. Johnson, and J. Leigh, TeraVision : a distributed, scalable, high resolution graphics streaming system, Proc. IEEE International Conference on Cluster Computing (CLUSTER '04), pp.391-400, 2004.
- [38] L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, C. Zhang, G. Goldman, J. Leigh, A. Johnson, SAGE : the scalable adaptive graphics environment, Proc. Workshop on Advanced Collaborative Environments (WACE 2004), 2004.
- [39] N. K. Krishnaprasad, V. Vishwanath, S. Venkataraman, A. G. Rao, L. Renambot, J. Leigh, A. E. Johnson, and B. Davis, JuxtaView - a tool for interactive visualization of large imagery on scalable tiled displays, Proc. IEEE International Conference on Cluster Computing (CLUSTER '04), pp.411-420, 2004.
- [40] C. Zhang, J. Leigh, T. A. DeFanti, M. Mazzucco, and R. Grossman, TeraScope : distributed visual data mining of terascale data sets over photonic networks, Future Generation Computer Systems, vol.19, pp.935-

- 943, 2003.
- [41] A. Streit, R. Christie, and A. Boud, Understanding next-generation VR : classifying commodity clusters for immersive virtual reality, Proc. International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPH-ITE '04), pp.222-229, 2004.
 - [42] H. Chen, Y. Chen, A. Finkelstein, T. A. Funkhouser, K. Li, Z. Liu, R. Samanta, G. Wallace, Data distribution strategies for high-resolution displays, Computers & Graphics, vol.25, pp.811-818, 2001.
 - [43] Y. Chen, H. Chen, D. W. Clark, Z. Liu, G. Wallace, K. Li, Software environments for cluster-based display systems, Proc. IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '01), pp.202-211, 2001.
 - [44] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahern, P. D. Kirchner, and J. T. Klosowski, Chromium : a stream-processing framework for interactive rendering on clusters, Proc. ACM SIGGRAPH, pp.693-702, 2002.
 - [45] M. Roth, G. Voss, and D. Reiners, Multi-threading and clustering for scene graph systems, Computers & Graphics, vol.28, pp.63-66, 2004.
 - [46] D. Pape, C. Cruz-Neira, and M. Czernuszenko, CAVE user's guide, Electronic Visualization Laboratory, University of Illinois at Chicago, 1997.
 - [47] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira, VR Juggler : A virtual platform for virtual reality application development, Proc. IEEE Conference on Virtual Reality (VR '01), pp.89-96, 2001.
 - [48] B. Schaeffer and C. Goudeseune, Syzygy : native PC cluster VR, Proc. IEEE Conference of Virtual Reality, (VR '03), pp.15-22, 2003.
 - [49] N. Hashimoto, Y. Ishida, and M. Sato, A self-distributing software environment for immersive multi-projector displays, Systems and Computers in Japan, vol.38, pp.1-9, 2007.
 - [50] H. Peng, H. Xiong, and J. Shi, Parallel-SG : research of parallel graphics rendering system on PC-Cluster, Proc. ACM international Conference on Virtual Reality Continuum and its Applications in Industry (VR-CAI '06), pp.27-33, 2006.
 - [51] T. Ogi and T. Uchino, Dynamic load-balanced rendering for a CAVE system, Proc. ACM Symposium on Virtual Reality Software and Technology (VRST '06), pp.189-192, 2006.
 - [52] J. Jacobson, and M. Lewis, Game engine virtual reality with CaveUT, IEEE Computer, vol.38, pp.79-82, 2005.
 - [53] A. Juarez, W. Schonenberg, and C. Bartneck, Implementing a low-cost CAVE system using the CryEngine2, Entertainment Computing, vol.1, pp.157-164, 2010.
 - [54] T. Schou and H. J. Gardner, A Wii remote, a game engine, five sensor bars and a virtual reality theatre, Proc. Australasian Conference on Computer-Human Interaction : Entertaining User Interfaces (OZCHI '07), pp.231-234, 2007.
 - [55] OptiTrack, NaturalPoint, <http://www.naturalpoint.com/optitrack/>
 - [56] R. M. Taylor, II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, VRPN : a device-independent, network-transparent VR peripheral system, Proc. ACM symposium on Virtual Reality Software and Technology (VRST '01), pp.55-61, 2001.
 - [57] VRPN, <http://www.cs.unc.edu/Research/vrpn/>
 - [58] XML CppDom, <http://sourceforge.net/projects/xml-cppdom/>
 - [59] Generic Geaphics Toolkit, <http://ggt.sourceforge.net/>
 - [60] Boost C++ Libraries, <http://www.boost.org/>
 - [61] CentOS, <http://www.centos.org/>
 - [62] 目野大輔, 陰山聡, VR Jugglerによる3次元可視化, スーパーコンピューティングニュース, vol.13, pp.34-45, 2011.

9. 付録

VR Jugglerのインストールには、一連の依存ライブラリを予めインストールしておく必要がある。しかし、日本語の文献や情報が少ないため、特にLinuxにインストールする場合は、ある程度の知識がないと、インストールに手間取ると考えられる。そこで、VR Jugglerのビルド環境の構築に必要な作業を、以下に示すようなビルド・スクリプトとして用意した。以下の操作は、root permissionの下でBorne shellを使って行う。

install flagpoll
ソフトウェアのコンパイルに必要な情報を収集するツール

```
tar xfv ../flagpoll-0.9.4-src.tar.bz2
cd flagpoll-0.9.4-src/
./setup.py install
```

install scones
ソフトウェアのビルド・ツール

```
tar xfv ../scons-2.1.0.tar.gz
cd scons-2.1.0
python setup.py install
```

install boost
C++の標準ライブラリ

```
tar xfv ../boost_1_48_0.tar.gz
cd boost_1_48_0
./bootstrap.sh
./b2
./b2 install
```

install CppDom
C++用DOM解析ツール

```
tar xfv ../cppdom-1.0.3.tar.gz
cd cppdom-1.0.3
scons BoostBaseDir=/usr/local/ BoostIncludeDir=/usr/local/include prefix=/usr/local var_arch=x64
scons install
```

install gmtl
一般数学テンプレート・ライブラリ

```
tar xfv ../gmtl-0.6.1.tar.gz
cd gmtl-0.6.1
scons
scons install
```

install VRPN
VR周辺機器をネットワーク経由で接続するためのクラス・ライブラリ

```
unzip ../vrpn_07_29.zip
cd vrpn
mkdir build
cd build
cmake ../.
make -j4
make install
```

※cmakeをインストールしておく必要がある

```
# install vr-juggler
# VRJuggler本体
```

```
LD_LIBRARY_PATH=/usr/local/lib:/usr/local/lib64:/usr/local/share
export LD_LIBRARY_PATH
tar xfv ../vrjuggler-3.0.1-1-src.tar.bz2
cd vrjuggler-3.0.1-1-src
mkdir build.linux.posix
cd build.linux.posix
../configure.pl --with-boost=/usr/local/ --with-vrpn=/usr/local
gmake build
gmake install
```

```
# install Doozer
# コンパイル処理を単純化するためのツール
```

```
tar xfv ../Doozer-2.1.6.tar.gz
cd Doozer-2.1.6
./configure install
make
make install
```

表A configuration fileの設定パラメータの例

display_system	Value	surface_viewports	Value
number_of_pipes	1	origin	0.15000, 0.01172
pipes	0.0	size	0.78125, 0.97656
display_windows	Value	view	Left Eye
origin	0, 0	lower_left_corner	- 1.500
Size	100, 100	lower_left_corner	0.000
Pipe	0	lower_left_corner	- 1.5000
stereo	false	lower_right_corner	1.500
border	false	lower_right_corner	0.000
hide_mouse	true	lower_right_corner	- 1.500
full_screen	true	upper_right_corner	1.500
always_on_top	false	upper_right_corner	3.000
active	true	upper_right_corner	- 1.500
allow_mouse_locking	true	upper_left_corner	- 1.500
start_locked	KEY_NONE	upper_left_corner	3.000
start_locked	false	upper_left_corner	- 1.500
sleep_time	75	user	Observer
		active	true
		tracked	false

(2013年11月12日受理)