

平成18年度
研究開発成果報告書

テレ・イマーシブ・カンファレンス・
システムに関する研究

委託先： (株)ケイ・ジー・ティー

平成19年4月

情報通信研究機構

平成18年度 研究開発成果報告書 (一般型)

「テレ・イマーシブ・カンファレンス・システムに関する研究」

目次

1	研究開発課題の背景	3
2	研究開発の全体計画	5
2-1	研究開発課題の概要	5
2-2	研究開発目標	5
2-2-1	最終目標	5
2-2-2	中間目標	5
2-3	研究開発の年度別計画	7
3	研究開発体制	8
3-1	研究開発実施体制	8
4	研究開発実施状況	
4-1	IPT(Immersive Projection Technology)用VR基盤ライブラリの研究開発	9
4-1-1	序論	9
4-1-2	OpenCABIN libraryの詳細設計と開発	9
4-1-3	3次元データフュージョンの研究	12
4-1-4	まとめ	16
4-2	空間共有アプリケーション構築用ライブラリの研究開発の研究開発	17
4-2-1	序論	17
4-2-2	空間共有ライブラリの開発	17
4-2-3	通信基盤システムの実装	20
4-2-3-1	VPNの実装	20
4-2-3-2	UDP通信の導入	21
4-2-3-3	通信ライブラリ	22
4-2-3-4	パフォーマンス測定	24
4-2-4	通信基盤ライブラリと既存ソフトウェアの移植性調査	26
4-2-5	まとめ	29
4-3	知識創造プロセスの支援のためのデータベース開発	30
4-3-1	序論	30
4-3-2	タイル型ディスプレイ向けKJ法システムの開発	30
4-3-3	タイル型ディスプレイを用いた遠隔コミュニケーションにおける アイコンタクトの評価実験	33
4-3-4	まとめ	34
4-4	アドバイザー・グループの需要研究	35
4-4-1	序論	35
4-4-2	遠隔授業と研究開発者のコラボレーションと実証試験	35
4-4-3	マルチ画面システムを使った空間共有システムの開発と実装試験	36

4-4-4	高等教育支援システムの市場機会の検討.....	39
4-4-5	アドバイザー・グループとの連携によるプロトタイプ開発	41
4-4-6	N3VR 研究会における CnC 基盤ライブラリの啓蒙活動	42
4-4-7	まとめ.....	42
4-5	総括	43
5	参考資料・参考文献.....	44
5-1	研究発表・講演等一覧.....	44

添付資料

<別紙 1> S C 2006 出張報告

<別紙 2> K J 法システム構成とクイックスタートガイド

1 研究開発課題の背景

(1) 既存コミュニケーションツールの意思疎通力や臨場感の不足

ネットワークの発達によって、今日の社会では遠隔地の利用者と協調して意思決定や共同作業を行うことが多くなってきた。このような遠隔協調作業を行うためには、E-mail や WWW ブラウザ、TV 会議システム等を使用し、マルチメディア情報を活用したコミュニケーションが有効的である。しかしながら、これらのシステムでは扱える情報が限られ、感情が伝わらないなどの制約が多く、実空間でのコミュニケーションの補佐的役割でしか使用されていないのが現状である。特に、遠隔授業や研究開発、プロダクトデザインなどの分野では対象に三次元表現が必要となる場合も多く、現行の遠隔コミュニケーションシステムでは、意思疎通力や臨場感不足という問題がある。

(2) IPT に求められる知識発見、知識創造支援機能

一方、全国各地に点在している CAVE や CABIN に代表される IPT(Immersive Projection Technology)をネットワークで接続し、没入型の VR 空間を共有しようという試みが行われている。これらの研究では、お互いが**アバタ**（代理人）と呼ばれる CG のキャラクターや、カメラで撮影されたユーザの映像を相手の VR システムに投影することによってコミュニケーションを実現する。またお互いの VR 空間に同じオブジェクトデータ（物体）を表示することによって、あたかも同じ空間を共有しているような高い臨場感を生成する。これらの高臨場感のある協調作業では、お互いの利用者が知恵や知識を共有し、膨大なデータの中から有益な情報を抽出し、問題解決にあたるデータマイニング等の知識発見や知識創造支援機能が重要な要素となっている。IPT は、大画面表示機能があるので、多様なデータやコンテンツを一覧できるので、知識創造支援機能を本質的に有しているといえる。

(3) IPT に必要な互換性

共有 VR 空間に関する研究はいくつかのプロジェクトで研究され、ある一定の成果を収めてはいるが、いずれもプロジェクト間での互換性が乏しいという問題がある。例えば、あるグループが開発した共有 VR 空間と、他のグループが開発した共有 VR 空間を接続することは通常困難であり、現状では閉鎖されたグループの中でしか利用することができない。この原因としては、システム間の互換性の低いハードウェアと、個々のシステムが異なるライブラリを用いているというソフトウェアの要因が考えられる。IPT のシステムは非常に高価であるがゆえ、通信相手ごとに新規のシステムを導入することは困難であり、新たなライブラリやフレームワークを導入することも、これまでに蓄積されてきたソフトウェア資産を捨てることになり、普及の妨げとなっている。

(4) IPT に必要なアプリケーション・ソフトウェア

VR 装置の本来的な価値は、意思疎通力に優れ、臨場感があり、多様なデータを一覧的に表示でき、コミュニケーション力と知識創造支援力を有していることである。しかしながら、この特長を生かして、普段利用しているアプリケーション・ソフトを IPT でも利用したいと思っても、極めて困難である。これは、アプリケーション・ソフトウェアが、IPT で利用されることを想定されていないからである。

その理由は、IPT 機能を組み込むコストが高いからである。組み込み用のライブラリは、非常に高価であるし、情報もないからである。もし、費用をかけずに、簡単に組み込みができれば、IPT 対応のアプリケーション・ソフトウェアは、増大すると思われる。なぜなら、普段利用しているソフトウェアのプルダウンメニューの中に、「IPT 出力」や「ネット会議の実行」等のボタンが埋め込まれていたら、ユーザにとっては、非常に便利な機能であり、気軽に利用ができるようになるからである。例えば、現在でも、マイクロソフト社のネット・ミーティングを利用する人が少ないのは、わざわざ別アプリケーションを立ち上げなければならないからと考えられる。研究者らの経験では、「素人度が高いほど、一つのアプリケーションで済ませたい」というのが、現実であるからである。

(5) 研究開発分野の現状

外部環境の変化として、テレ・イマーシブ・カンファレンス・システムを構築するハードウェアシステムの価格は、継続的に低下し続けている。特に高性能コンピュータ・グラフィックスシステムについては、シリコングラフィックスが開発を中止したため、ますます、PC向けの低価格グラフィックスカードや、それを用いた並列システムが台頭すると思われる。表示装置は液晶、プロジェクタとも高性能・低価格化が進んでおり、こちらでも1台の高価な表示装置ではなく、マルチディスプレイの活用が進みつつある。

ネットワークの高速化も進み、音声通信ではスカイプのようなビジネスモデルも登場している。このように、現状のTV会議システムを超えたテレ・イマーシブ・カンファレンスを実施するためのハードウェア環境は実現の方向へ加速していると思われる。

一方、社会環境の変化として、企業合併、企業内の横連携、アウトソーシングによる企業間連携など、連携の機会は増大している。医療分野に見られるような専門化に伴う、医師間連携は、一般の研究分野・設計分野にも広がる可能性を持っている。

さらに、仕事と家庭の両立のための在宅勤務や地方活性化のための地方サテライトオフィスなどでも、テレ・イマーシブ・カンファレンスの需要は期待できる。

次に競合技術については、立体視による没入感システムの研究開発は米国では少なくなり、マルチディスプレイに焦点が移動している。そのため、マルチディスプレイだけを対象としたシステムとしては米国の先進研究に優位性を譲るが、スケーラブル(両者の対応)な環境構築という点で優位性はある。

また、ソースコードの改変無くソフトウェアを連携させる思想は、米国のChromiumのような類似システムで採用されている。他、PDF3DでもOpenGLのキャプチャリング技術が使われている。まったく別分野だが、数値計算ソフトウェアのメモリ領域からデータを取り出す基盤ライブラリもJAXAで開発されており、リコンパイル・リリンクなしでのアプリケーション連携は、速いサイクルでのシステム開発に重要な技術として、今後も発展していくことが予想される。既にデジタル情報は溢れるほどに蓄積されつつあり、さらに、今後生成される。これらを合成するニーズは高まるに違いない。

2 研究開発の全体計画

2-1 研究開発課題の概要

本研究では、VR空間を共有しながら人間相互の意思疎通や協調作業を行うためのコミュニケーション環境を構築するためのソフトウェア・ライブラリを提供し、遠隔地間での協調的なデータマイニングやナレッジマネジメントを実現することを目指している。

本研究の課題は、

- ・ データベースを介在させ、「空間、時間、人物、物体、操作、情報の共有化」を有機的に、他のアプリケーション・ソフトウェアと連携利用できる事、
- ・ TV会議以上のスケーラビリティを有し、実用レベルまで実装することができる事、
- ・ 両方のライブラリが他のアプリケーション・ソフトウェアの開発基盤に資することができる、汎用性、オープン性、互換性を実現する事、である。

具体的な研究開発課題のサブテーマとしては、(1) IPT用VR基盤ソフトウェアの開発、(2) 空間共有アプリケーション構築用ライブラリの開発、(3) 知識創造プロセス支援のためのデータベースの開発、(4) 空間共有会議システムのプロトタイプ構築とその評価を行う。

2-2 研究開発目標

2-2-1 最終目標（平成20年8月末）

1) VR基盤ソフトウェア

デスクトップ環境からIPT環境まで同じAPIで、設定ファイルを変更することによって対応できるVR基盤ソフトウェアを整理する。コンソーシアムを設立し一般公開を行う。

2) 携帯端末インターフェイス

GUI評価を行い、VR空間に提示される3次元GUIより使いやすいように改良を行う。

3) グラフィカル開発環境

サンプル・プログラムとヘルプ機能を参照しながら利用者が独自のアプリケーションを構築できる環境と、モジュールの整備。

(2) 空間共有アプリケーション構築用ライブラリの開発

空間共有アプリケーション構築のための、空間共有機能、時間共有機能、人物共有機能、操作共有機能、情報共有機能、物体共有機能をライブラリとして整理する。これらのライブラリは、IPT用VR基盤ソフトウェアを始め、CABIN Lib.等の各種のVR構築用ライブラリとの併用を実現する。

またコンソーシアムを設立し一般公開を行う。

(3) 知識創造プロセスの支援のためのデータベースの開発

情報の相互影響を考慮し、関連情報の変化に対応しながら、情報のリンク関係を動的にグラフィカル提示する機能を組み込む。

(4) 空間共有会議システムの構築と評価

アドバイザーグループと連携し、プロトタイプの機能実証。オーグメンテッド・リアリティ・プロトタイプを試作し、機能の評価を行う。

2-2-2 中間目標（平成19年1月末）

(1) IPT用VR基盤ソフトウェアの開発

1) VR基盤ソフトウェア

デスクトップ版と1面スクリーン対応版を試作し、実空間の3次元知覚と差のない視覚情報を提示できる。多面スクリーンに用いた場合にスクリーンのつなぎ目に違和

感がないようにする。入力デバイスからのデータを管理するデーモンの試作。国際化のための仕様の決定。

2) 携帯端末インタフェース

GUI を用いたインタラクションを VR 空間で実現するために PDA と携帯電話をベースとしたユーザインタフェースを開発する。設定ファイルによって GUI が自動的に再構築できるようにする。

3) グラフィカル開発環境

ネットワークエディタの設計。グラフィックス関連、インタラクション関連、割り込み処理関連、デバイス関連の各モジュール群を構築する。

(2) 空間共有アプリケーション構築用ライブラリの開発

空間共有アプリケーションの構築に必要な、空間共有機能、時間共有機能、人物共有機能、操作共有機能、情報共有機能を、物体共有機能をそれぞれ実装し機能評価を行う。たとえば、人物の共有機能に関しては、アバタによる指差し位置がセンサ誤差と同等 (5cm 以内) の精度で伝達されること、時間の共有に関してはビデオ映像の 2 コマ (0.06 秒) 以内の時間同期を実現する、また情報の共有機能では SQL を介した種々のデータベースへのアクセス機能を実現することを目標とする。

(3) 知識創造プロセスの支援のためのデータベースの開発

データベースに蓄積された情報から利用者が求めるデータを的確に抽出するデータマイニング機能を開発する。登録するデータに属性を持たせ、関連ある情報を自動的にリンクする機能を持たせること。

(4) 空間共有会議システムの構築と評価

アドバイザーグループと連携し、VR タイプ、デスクトップタイプの仕様の決定。

2-3 研究開発の年度別計画

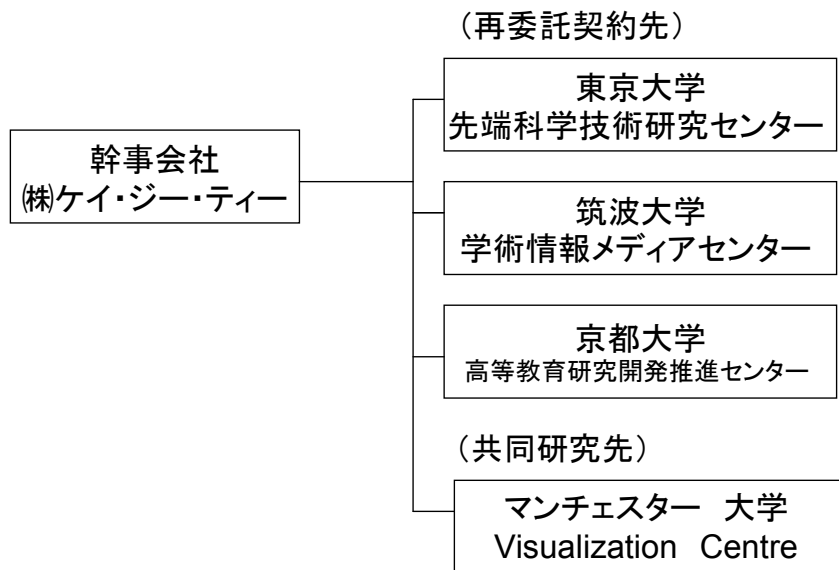
金額は非公表

研究開発項目	H16年度	H17年度	H18年度	H19年度	H20年度	計	備考
(1) IPT用VR基盤ソフトウェアの開発					→		東京大学
(2) 空間共有アプリケーション構築用ライブラリ の開発					→		筑波大学
(3) 知識創造プロセスの支援のためのデータベ ースの開発					→		(H17から) 京都大学
(4) 空間共有会議システムの構築と評価					→		
間接経費額（税込み）							
合 計							

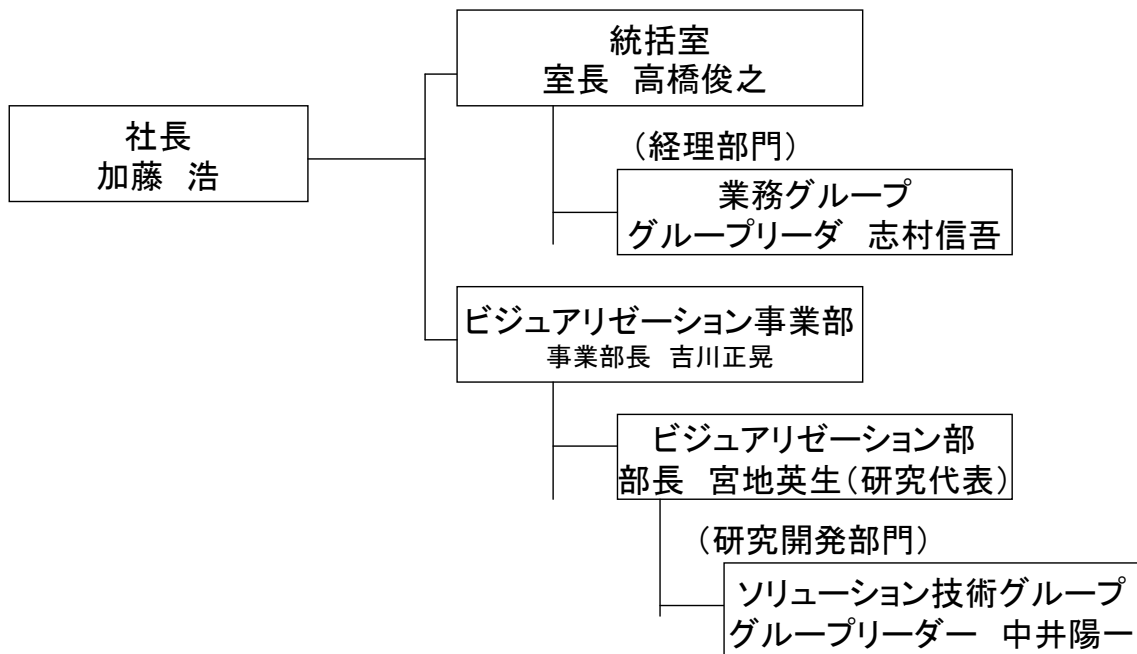
- 注) 1 経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の30%を上限として計上（消費税を含む。）。
- 2 備考欄に再委託先機関名を記載
- 3 年度の欄は研究開発期間の当初年度から記載。

3 研究開発体制

3-1 研究開発実施体制



受託機関における経理部門の体制と研究開発部門の体制



4 研究開発実施状況

平成 17 年度、京都大学-筑波大学-東京大学の IPT 施設を JGN2 で接続した 3 拠点通信試験を実施し、基本的な設計とプロトタイプ実装は終了した。平成 18 年度は、これらソフトウェアを公開用に整備し、一部、ダウンロードサービスを開始した。また、アドバイザー・グループの要求、プロトタイプシステム開発に対応するために追加開発を行った。

知識創造プロセスの支援のためのデータベースの開発では、昨年度開発した K J 法アプリケーションの改良とアイコンタクトに関する研究調査を行った。

空間共有会議システムの構築においては、遠隔授業実証試験、東京・大阪 TV 会議システム接続試験を実施した。また、アドバイザー・グループと共同で建築分野向けの研究開発 PR 用のプロトタイプシステムを開発した。

4-1 IPT(Immersive Projection Technology)用 VR 基盤ライブラリの研究開発

4-1-1 序論

このサブテーマでは、大きく 2 つの研究開発を進めている。1 つは、IPT システム、タイル型ディスプレイ用の基盤ライブラリ(OpenCABIN library)の開発、もう 1 つは市販アプリケーションとの容易な連携を図るための OpenGL フェュージョン技術の開発である。

OpenCABIN library においては、本年度、ソースコード一般公開のための最大の障害であった著作権問題を解決する目的、及び CABIN ライブラリプログラマからの旧来からの要望を満たすべく、再設計と実装する作業を行った。また、その成果物である第 0 版、第 1 版を一般公開し、普及のための準備を進めた。

OpenGL フェュージョン技術に関しては、性能調査の実施と改善、市販アプリケーションとの接続試験を実施した。また、既存製品に OpenGL フェュージョン技術を組み込み、フェュージョン VR として製品化し、販売を開始した。

4-1-2 OpenCABIN library の詳細設計と開発

(1) 本ライブラリ開発の意義

テレ・イマーシブ・カンファレンス・システムの重要な構成要素として、大画面ディスプレイが挙げられる。CABIN、CAVE に代表される没入ディスプレイは、高品位な仮想 3 次元空間を提示する。また、タイル型ディスプレイは多数のディスプレイを平面あるいは曲面状に配置して一つの大画面ディスプレイを比較的安価に構成できる。従来は、これらの大画面ディスプレイシステムの実現には高価なグラフィックワークステーションが必要であった。しかし、近年のギガビットイーサネットの普及、PC のグラフィックハードウェアの高性能化により、グラフィックワークステーションを導入しなくても、複数の PC をギガビットイーサネットで結び、一つのディスプレイシステムとして連動させる事がハードウェアの面では可能になった。これにより、ハードウェア導入コストをかなり押さえる事ができた。しかしソフトウェアの面では、スクリーン間情報同期問題を解決しなければならない。旧来のハードウェア共有メモリを前提としたソフトウェアを PC クラスタシステムで問題なくエミュレートするのは非常に困難である。そこで、本プロジェクト開発では新ライブラリを新規作成することにした。PC クラスタ型を前提としたマルチスクリーンシステムを前提とした情報共有 API を新たに設ける事により、開発効率と実行効率の両方を狙う事にした。

さらに、本プロジェクトの目的であるテレ・イマーシブ・カンファレンス・システムの実現・普及には、大画面ディスプレイを複数サイトで同調させる必要がある。サイトによって大画面ディスプレイシステムの構成は大きく異なることが予想される。あるサイトでは、没入ディスプレイを使用し、また別のサイトでは、タイル型ディスプレイを持つサイトの参加もあるだろう。よって、テレ・イマーシブ・カンファレンス・システムのアプリ

ケーションのための基盤ソフトウェアは、大画面ディスプレイシステムの相違点を吸収できるように設計されていなければならない。OpenCABIN library では、没入ディスプレイやタイル型ディスプレイなど、ディスプレイシステムの違いを吸収し、アプリケーションプログラムを何処でも動作させる事ができるように設計した。

(2) 本ライブラリ実現のための検証実装

入力デバイスからのデータを処理するデーモンについては、操作系と位置センサ系の2種類がある。まず、操作系については、市販のあらゆる USB ゲームコントローラを使えるよう、ボタンやスティックの割り当てを適切にカスタマイズして使用できるデーモンを開発した。また、位置センサについても、従来の Polhemus 社の Ultratrak だけでなく、Polhemus 社の Liberty や Ascension Technology 社の Flock of Bird のデータを処理するデーモンを試作した。glCABINlib2 での動作を確認している (図 4-1-1)。

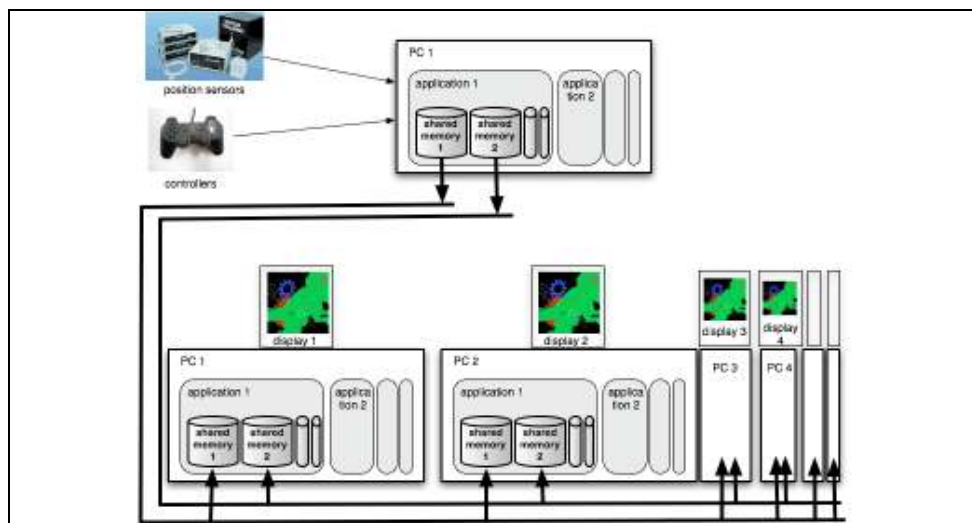


図 4-1-1 入力装置と情報共有機構：入力装置からの情報は、情報共有機構によりアプリケーションに伝達される。

また、大画面ディスプレイシステムでは Viewport と View Frustum の正しい設定が必要不可欠である。glCABINlib は CABIN 専用のライブラリであったため、これらはプログラム中にハードコーディングされていた。寸法についてはパラメータ化されていたので再コンパイルにより対応することとなっていた。実際 COSMOS や CAVE に対応する際には、再コンパイル作業を行っている。この設定部分は、テキストファイルにより設定されるよう設計する必要がある。また、没入ディスプレイだけでなく、タイル型ディスプレイなどにも対応できなければならない。本年度はテキストファイルに記述された設定を読み込み処理するモジュールの設計を行い、それを試作した。

(3) 本年度のライブラリ本体開発と公開

昨年度は、開発した glCABINlib2 システムによって、新情報共有システムの動作確認実験、プラグイン機構の動作確認実験、それらの CABIN での動作確認、実際のアプリケーションの動作確認を行ってきた。しかし、glCABINlib2 は、旧 glCABIN library を元に開発しており、著作権不詳なコードが混入していた。センサ情報配布部では、旧情報共有機構が残っていた。

また、没入ディスプレイユーザには、その用途に応じて最適なグラフィックライブラリ (OpenGL Performer 等) を選択したいという要望がある。しかし、今までの実装では、

描画部が分離されておらず、開発者は OpenGL でしか開発できなかった。このままでは OpenGL Performer や Open SceneGraph などのグラフィックライブラリに対応する事は困難であった。また、Orad DVG などの特殊なグラフィックハードウェア、さらには、自作の 3次元ディスプレイ装置などへの対応も困難が予想された。

本年度は、まず公開を第一優先目標とし、著作権不詳コードの撤廃、旧情報共有機構の撤廃を行い、それらを補完する部分の開発を行った。さらに、描画部を独立させ(図 4-1-2)、将来の OpenGL Performer や Open SceneGraph などのグラフィックライブラリ、Orad DVG や自作 3次元ディスプレイ装置への対応の容易さを確保した。また、公開に際し、本ソフトウェア名を OpenCABIN library と呼称する事にした。また、その成果物である第 0 版、第 1 版を Web 上で一般公開し、本ライブラリ普及のための準備を進めた。

これらには大画面ディスプレイを普通の PC の画面でエミュレートする機能が備わっている(図 4-1-3)。アプリケーション開発者は、このエミュレーション機能を使用することにより、没入ディスプレイなしでもアプリケーションが正しく動くかどうかの確認が容易になった。例えば、情報共有機構の使用が適切かどうかを容易に確認することができるようになった。

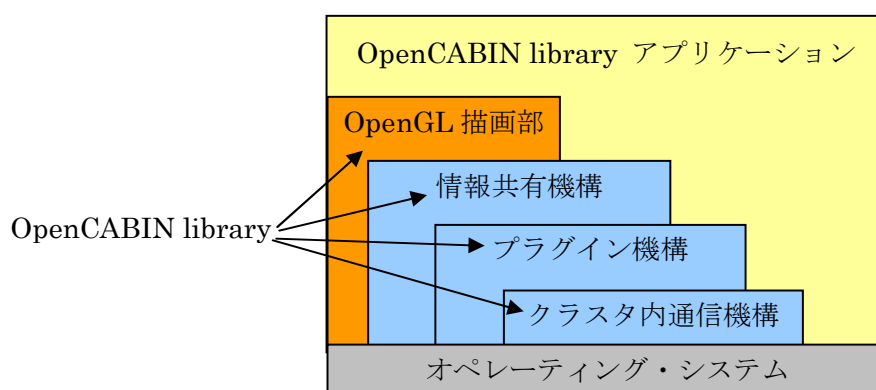


図 4-1-2 描画部を独立させた OpenCABIN library

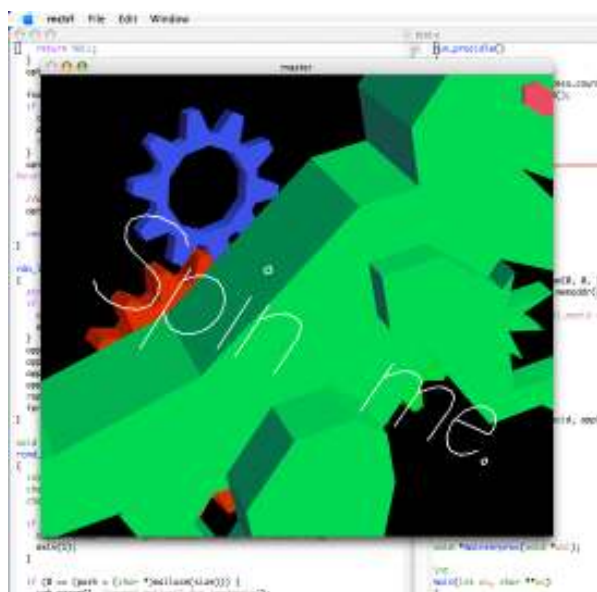


図 4-1-3 エミュレータ版の動作画面

4-1-3 3次元データフュージョンの研究

(1) ネットワーク透過型フュージョンの実装

当初の開発では、1台のWindowsのシステム上で複数のアプリケーションを1画面に合成表示していた。これを拡張して、OpenGL コマンドをネットワーク経由で複数のコンピュータに分配し、IPT 空間内での合成を可能にした。すでに、OpenGL アプリケーションを IPT 空間にコピーする技術は、(株)フィアラックスにより開発され「EasyVR」という名称で製品化されている。そこで、このソフトウェアのオプションの機能として OpenGL フュージョン技術を加え、FusionVR という名前で製品化した。

図 4-1-4 に拡張した実装部について示す。

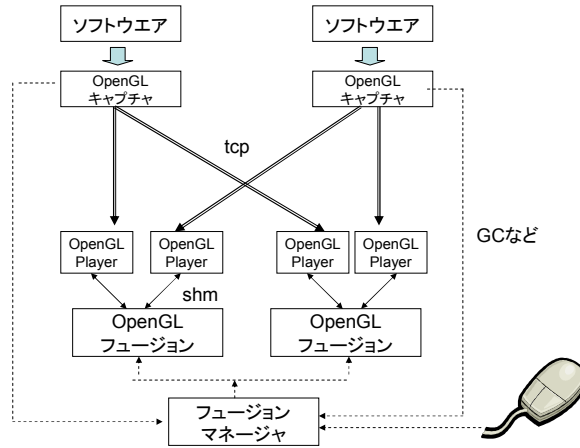


図 4-1-4 Fusion VR 構成図

ここでは CAVE™ のように、複数のスクリーンの各面に 1 台のコンピュータが画像を出力するタイプの IPT システムを想定している。表示用のコンピュータでは OpenGL Player というソフトウェアが、合成するソフトウェアの数だけ動作する。一方、OpenGL キャプチャは、キャプチャした OpenGL コマンドを、すべてのコンピュータで動作する OpenGL Player に対してコピーする。表示用コンピュータ上で、複数の OpenGL Player からの出力がフュージョンによって合成される点は、従来の方式（1 台内合成）と同じである。

複数の表示面間での同期を取るためにフュージョンマネージャが任意の 1 台のコンピュータ上で動作し、ここでグラフィックス・コンテキストを一括で管理する仕組みとした。

(2) OpenGL フュージョンの性能測定

・ 試験設備

ベンチマーク試験には 3 台の計算機を利用した。そのスペックを表 4-1-1 に示す。マシン間は fast イーサ(100Mbps)で接続されている。

表. 4-1-1 ベンチマーク試験用計算機環境

	CPU	Memory	Video Card	OS	Video Driver
PC1	Pentium4 2.8GHz	2GB	NVIDIA GeForce6600	Windows XP 2002SP2	84.21
	"	"	"	Fedora Core 4 (2.6.11-1369_FC4smp)	1.0-8756
PC2	Athron64 3500+ 2.2GHz	2GB	NVIDIA GeForce6600G	Windows XP 2002SP2	84.21
	"	"	"	Fedora Core 4 (2.6.11-1369_FC4)	1.0-8756
PC3	Athron64 3500+ 2.2GHz	2GB	NVIDIA GeForce6600G	Windows XP 2002SP2	84.21
	"	"	"	Fedora Core 4 (2.6.11-1369_FC4 x86_64)	1.0-8756



図 4-1-5 Atlantis1000 の表示例

ベンチマークに利用したプログラムはアトランティスを改変し、登場するサメを 4 匹にした atlantis4 と 1000 匹にした atlantis1000 (図 4-1-5) を使った。

・基本性能試験

まず、それぞれの計算機の基本性能を見るために Atlantis4 と Atlantis1000 を、1～4 プロセス、単体の計算機上で動作させた。OS の依存関係を調べるために、それぞれの計算機で Windows と Linux の両方を試験した。図 4-1-6 に Atlantis4 の結果を、図 4-1-7 に Atlantis1000 の結果を示す。

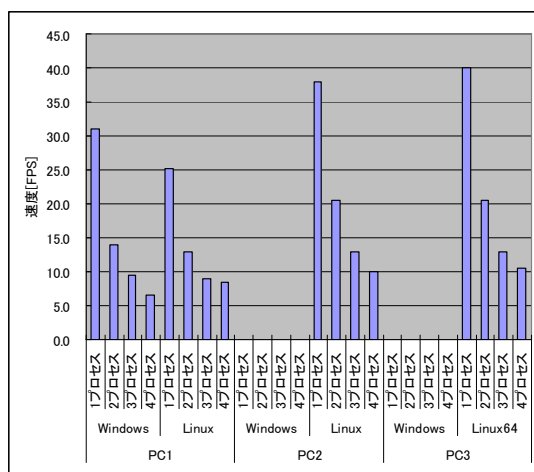
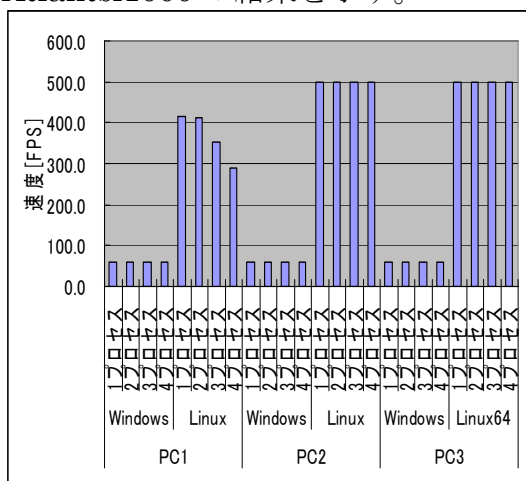


図 4-1-6 Atlantis4 のベンチマーク結果

図 4-1-7 Atlantis1000 のベンチマーク結果

図 4-1-6 より、Windows では、毎秒 60 回以上の表示の更新は実施しないのか、あるいは、計測できないことが分かる。そのため Atlantis1000 は、Linux を中心に計測を行った。図 4-1-7 では、この状態で、グラフィックス能力が飽和状態にあり、プログラム数が 2 倍になると表示性能[フレーム/秒]は半分になっている。

・ OpenGL キャプチャのオーバーヘッド

OpenGL キャプチャにおけるオーバーヘッドを調査するために、通信や合成をしないで、1 台のマシンで単体で動作させた場合と OpenGL キャプチャを実施した場合の性能を比較した。OpenGL キャプチャプログラムは、実際に通信はしないが、それまでの準備、例えば、OpenGL コマンドの解析、通信用のバッファメモリのアロケーションや、その領域へのデータのコピーは実施している。その結果をアトランティスを単体で動作させた場合の描画速度を 1 としたときの比率に変換したものを図 4-1-8 に示す。

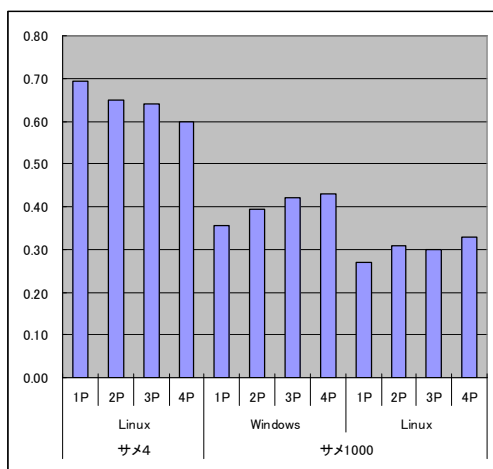


図 4-1-8 OpenGL キャプチャによる速度低下

サメが 4 匹の場合で 30～40%、サメが 1000 匹の場合には 60～75%の性能の低下が見られる。プロセス数と速度低下の関係は、サメ 4 匹では、プロセスが多くなるほど速度低下

率は高く、逆にサメ 1000 匹ではプロセス数が増えると、速度低下率はやや緩和されるようである。

・ 転送・表示のオーバーヘッド

次に PC 2 で OS を Linux にし、Atlantis1000 を動作させたときの、通信・合成表示をした状態（合成表示）、そして、合成後、オリジナルの Atlantis10000 を止めて OpenGL フェージョンが独自に表示を更新している状態（トラッキング時）の性能を測定した結果を図 4-1-9 に示す。

Atlantis を止めると、通信がまったく止まり、また、Atlantis 自身の表示も止まるので、そのときの性能は、OpenGL フェージョンの仕組みを使って、CAD のようなソフトウェアから 3 次元表示を VR 空間へコピーした後の、ヘッドトラッキングをしている状態での性能を表すことになる。

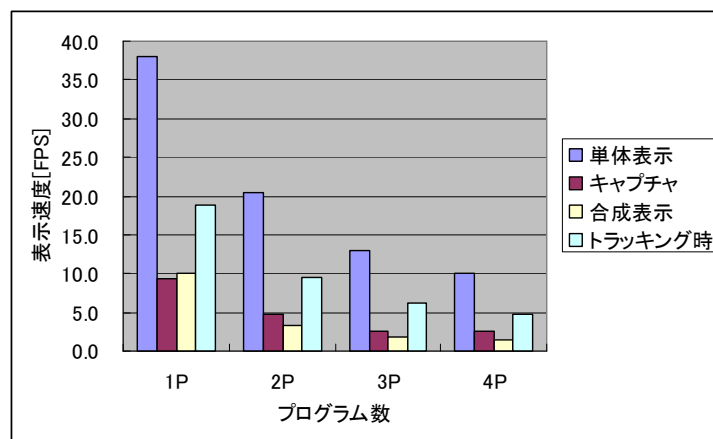


図 4-1-9 合成表示とヘッドトラッキング時の性能

この結果から、通信と合成には、それほど大きなオーバーヘッドは無いように見える。図 4-1-9 において合成表示の測定をしているときは、Atlantis の元々の表示と合成後の表示が同時に実施されていることになるので、図中の同じプログラム数における単体表示と直接比べるのではなく、1 P の合成表示は 2 P の単体性能と、2 P の合成表示は 4 P の単体表示と比較するべきである。

一方、Atlantis を止めてヘッドトラッキングの状態にした場合、オリジナルの Atlantis の表示が止まるので、サメの表示数は合成表示の半分に戻る。例えば、2 P の単体性能ではサメ 1000 匹を表示するプログラムが 2 本動作し、トラッキング時はサメ 2,000 匹を表示するプログラムが 1 本動作している状態になっている。

・ 高速化のための検討

OpenGL キャプチャのオーバーヘッド

OpenGL のキャプチャリングによるオーバーヘッドは試験前の予想に反して、意外に大きいことが判明した。ここでは、OpenGL コマンドの内容をチェックする。GC 関連はフェージョンマネージャーへ、それ以外は OpenGL Player へ転送する。また、転送は適宜バッファリングを行うので、そのメモリのアロケーションや、そこへのコピー処理も発生している。メモリアロケーションのロス無くすには、バッファを固定長にすれば良いが、それでは命令途中で通信が発生し、オリジナルのソフトウェアを通信の間、待たせることになる。決定的な改善策は無いがプログラミングによって、いくつかの改善は図れると思われる。

通信・合成表示のオーバーヘッド

通信は、かならず一定の時間がかかるのでオーバーヘッドの発生は避けられない。ここでは、複数のプログラムの合成を行うことによって、スケジューリングによる待ち時間が発生することを懸念していたが、今回のようにグラフィックスボトルネックが発生している状況においては、スケジューリングのオーバーヘッドは顕著ではないようだ。ただ、キャプチャまでの時間と合成表示を行ったときの時間を比較すると、少しずつプログラム数の増加に従って遅れが増えている。通信とスケジューリングのどちらが大きく寄与しているかは分からなかった。

ヘッドトラッキング時のオーバーヘッド

アバタプログラムや Atlantis のように、随時表示内容が更新される OpenGL プログラムは、本来、GL リプレースメントに向かないとされている。しかし、静止画のシーンの VR 空間への輸入機能は、本システムが本来得意とする領域である。しかし、今回のテストでは、表示負荷をほぼ同等にした場合、単体としてプログラムを動作させた場合に比べ、合成することによって性能が丁度半分になっていることが分かった。これは、2つの原因が考えられる。1つは、プログラムが複数動作するとき、GCのスイッチはグラフィックボードで行われるが、本システムではフュージョンマネージャーがソフトウェア的にスイッチ命令を送ることによる速度低下である。これは、さらに小さいプログラムをたくさん動作させたときには、より顕著になると考えられる。

もう1つはスクリーン間の表示同期を取るための遅延である。こちらは、スクリーンの数に制限を受けるので最大で6程度になる。

(3) ビデオアバタアニメーションとアニメーション合成

性能測定の後、いくらかのコーディング的な改良を加え、Atlantis のアニメーションとビデオアバタの合成試験を行った。ビデオアバタは、あらかじめ録画したビデオ画像と背景画像から、背景差分によって CG を生成するプログラムである。ビデオアバタの空間に、人が歩き回り、その空間をサメが泳ぎまわるといった不思議な光景を見ることができた(図 4-1-10)。

立体視の状態ではないが、別途実施した Atlantis とビデオアバタの2つのソフトウェア合成試験で毎秒 20 フレームという実用可能な範囲に性能が改善されていることを確認した。



図 4-1-10 IPT 空間でのソフトウェア合成立体視表示の様子

(3) シーングラフサポートの検討

OpenGL フェージョンの実用について検討していくと、正確な位置合わせが重要であることが解った。現状では、アプリケーション・ソフトウェアから送信されてくる OpenGL コマンドに含まれるカメラ情報とオブジェクト情報の区別が完全にできていない。そのため、個々のアプリケーションに対して、実際にフェージョンの試験を行い、カメラ情報の送信状況を調査して、個別対応しなければ正確な位置合わせができない。

図 4-1-11 は、建築分野向け研究成果 PR 用のアプリケーション構築例で、建築 CAD の情報と数値計算結果の可視化情報をフェージョン技術で統合した様子である。



図 4-1-11 建築 CAD 情報と室内気流の流れの合成表示例

この他、アドバイザー・グループから、いくつかの利用ニーズを吸い上げたが、多くは正確な位置表示が必要であり、今年度の事例開発では、その調整にかなりの工数が必要であった。これらの作業を通して、相当数のアプリケーション対応例が集まり、パターン分類することで調整を迅速にできるのではないかと、いう状況になっている。

また、表示内容の保存と再生という要求も出てきている。

これらの要求への対応にはシーングラフの導入が必要であると考えられる。

4-1-4 まとめ

OpenCABIN library は、公開用のコード整備が終了し、ダウンロードサービスを開始した。但し、ドキュメントの整備ができていないこともあり、利用者が居ない状況にある。今後はドキュメントの整備に注力し、サブテーマ 4 の普及活動を強化していく予定である。

OpenGL フェージョン技術は、詳細な速度調査を実施し、一定の改良できた。しかし、アドバイザー・グループとのプロトタイプ共同開発において、新しく正確な位置あわせのニーズが浮上した。今年度はプロトタイプ開発のために位置合わせ機能を実装したが、今後は汎用的に位置あわせを可能とする仕組みを提供する必要がある。

また、OpenCABIN library と OpenGL フェージョンを同時に使う場合についても、実証試験を行い、改良を加える予定である。

4-2 空間共有アプリケーション構築用ライブラリの研究開発

4-2-1 序論

空間共有アプリケーション構築用ライブラリは、仮想空間を遠隔地間で共有する機能実装を支援するための空間共有ライブラリ（東京大学：MVL Toolkit の普及用整備と拡張）と、通信プログラム開発支援のための通信基盤システムの開発から成る。

平成 18 年度、空間共有ライブラリは遠隔授業の実証試験と平行して、試験評価を行いながらライブラリ整備を実施した。通信基盤システムは、UDP 実装、ファイヤウォール越え、API 設計を行い、音声通信ソフトウェアへの移植作業を実施した。

4-2-2 空間共有ライブラリの開発

(1) 実装状況

仮想空間を遠隔地間で共有するために必要な機能要素を、空間共有機能、時間共有機能、人物共有機能、操作共有機能、情報共有機能、物体共有機能に整理し、それぞれの実装を行った。

空間の共有は、サイト間で利用者の位置情報を相互に交換し同一のプログラムを動かすことで、お互いの位置関係を表示した空間共有を実現する。また、アニメーションやビデオ再生時における時間の共有機能は、起動時やコマンド送信時における時間合わせの機能と、プログラム内でのレンダリング時間を考慮した時間制御機能によって実現する。人物の共有機能に関しては、人物の撮影映像をキャプチャし背景差分を行った人物映像をサイト間で送受信し、仮想空間に合成するビデオアバタ機能の開発を行った。操作の共有は仮想空間におけるインタラクション情報を送受信することでサイト間での整合性を保つ機能である。ここではコントローラの位置と操作データを送受信することで、仮想指し棒、ポインタ、仮想キーボード等のインタフェース機能の実装を行った。また、情報の共有機能は、ODBC を介して MySQL 等で構築したデータベースへのインタフェースを構築し、画像データ、3次元 CG モデル、PowerPoint ファイル等を格納、検索する機能を構築した。これを前述の操作共有機能と組み合わせることで、各サイトからネットワーク上のデータベースにアクセスし、必要な情報を共有することができる。また情報共有のためのホワイトボード機能を実装した。これは仮想キーボードやポインタを使用して文字や絵を記述ことができ、ホワイトボード上の情報は遠隔地間で共有することができる。物体共有機能としては、ステレオカメラで撮影された実物体や背景の映像をリアルタイムで共有仮想空間に合成することで、CG 映像だけではなく実映像を用いて情報を共有する機能を構築した。

これらの機能は、ライブラリとして整備が行われ次第、公開していく予定である。

(2) ビデオアバタ（人物共有機能）を用いた通信テスト

本研究では通信基盤として JGN2 (Japan Gigabit Network 2) とつくば WAN を使用したネットワーク実験環境を構築し、基本となるビデオアバタ機能を用いた多地点間の通信実験を行った。使用した没入型ディスプレイは、筑波大学の CS Gallery、東京大学の CABIN、京都大学の CAVE であり、各ディスプレイは 3 面のスクリーン構成で使用した。各サイトでは、通信用 PC (Red Hat Enterprise Linux WS V4 (EM64T) Intel Xeon 3.0GHz) を配置し、それに 3 面スクリーンに対する 3 台ずつの映像描画用クラスタ PC を接続した。撮影環境としては SONY DFW-X710 の IEEE1394 カメラを通信用 PC に取り付け、リアルタイムで撮影および通信を行った。ビデオカメラで撮影される映像は解像度 640×480 ピクセルの RGB 画像とし、毎秒 15 フレームでキャプチャを行った。

図 4-2-1 は 3 地点間でビデオアバタを用いたコミュニケーションを行っている様子を示したもので、また表 4-2-1、表 4-2-2 はそれぞれの地点間での映像転送速度と各没入型ディスプレイ内でのビデオアバタの描画フレームレートを示したものである。表 4-2-1 より各サイト間で、約 15Hz の通信が実現されており、カメラからのキャプチャ速度に対してデー

タ損失の無い通信を実現できていることが示された。また、各没入型ディスプレイ内での描画速度には差があったが、これは描画用 PC のレンダリング性能の影響を受けているものと思われる。



図 4-2-1. 3 地点間でのビデオアバタ通信実験

表 4-2-1. 3 地点間でのビデオアバタ映像の伝送速度

	筑波大学	東京大学	京都大学
筑波大学	-	15.0	15.0
東京大学	14.9	-	14.9
京都大学	14.9	15.0	-

表 4-2-2. 各没入型ディスプレイ内での描画速度

	筑波大学	東京大学	京都大学
描画速度	15.3	7.15	9.97

(3) 操作共有と情報共有機能の実装と試験

次に、没入型ディスプレイ間で空間共有を行いながら、操作共有、情報共有機能等の機能について検証を行うための実験を行った。実験環境としては、筑波大学の CS Gallery と JGNII つくばリサーチセンターの没入型拡張現実感ディスプレイ AR View 間のつくば WAN のネットワーク環境を使用した。

操作共有機能としては、コントローラを用いたポインタ、指し棒機能を実装した。ポインタはコントローラ位置にカーソルを表示し、利用者の手先の位置を伝達する機能であり、指し棒は手の位置からレーザポインタ状の指し棒を表示し差し位置を伝達する機能である。ポインタと指し棒の位置情報伝達について実験時にアンケートを行った結果から、ポインタにより指先位置を示す方法よりも指し棒によって対象を指し示す方法の方が分かりやすいとの意見が得られた。

また情報共有機能に関しては、MySQL のデータベースに 3 次元 CG モデル、画像データ、PowerPoint スライド等のデータを格納し、共有仮想空間で検索を行い遠隔地間でデータを共有できることを確かめた。また仮想キーボードとポインタカーソルを用いたホワイトボード機能の実装を行った。これは共有空間に表示されたホワイトボードに仮想キーボードを用いた文字出力やポインタカーソルによるフリーハンドの描画を行い情報を共有する機能であり、遠隔地間でのメモ情報の共有等に利用できることが確かめられた。

また物体共有機能に関しては、一方のサイトにステレオカメラを設置し、背景のステレオ映像を伝送しながら、ビデオアバタ、3 次元データとの合成を行った。背景と人物像は共に実写映像を利用しているが、別々に伝送が行われるため、人物像と背景の間に CG モデルを置く等の自由な位置関係表現を行うことができた。

下図はホワイトボードによる情報共有(図 4-2-2)、ステレオカメラによる実写映像空間

の共有(図 4-2-3)を行っている様子を示したものである。



図 4-2-2. ホワイトボードを使用した情報共有



図 4-2-3. ステレオカメラによる実映像空間の共有

4-2-3 通信基盤システムの実装

平成 17 年度の通信サーバ接続実験より、以下の課題が明らかになった。平成 18 年度はこれらを新たな課題と捉え、調査、研究、実装、実験を進めてきた。

(1) ファイアウォール超え

「安全な通信路の確保」のため、昨年度は PPTP を使い VR 端末と通信サーバ間は VPN 接続とした。しかしながら、VR 端末が設置される各組織によっては、このパケットを通さない設定にしているところがあり、広く普及させるためには、より一般的な方法での通信路設定を行う必要がある。今年度は、ほとんどの組織のファイアウォールの設定で HTTP 80/TCP (TCP 80 番ポートを使った HTTP 通信)は通過許可していることから、そのポート番号を使って、VPN もしくは Tunnel 通信路を確立できる openVPN を検討し、実装した通信サーバを構築した。

(2) 効率的な通信

通信サーバと VR 端末との通信をすべて TCP パケットで行うと、確実にパケットは到着するが、どこか遅い回線があると、その速度で全体の通信性能が頭打ちになってしまう。

UDP パケットはその到着を待たずに次々とパケットを送出できるよう設計されているので、映像や音声等途中でデータ損失があっても通信路を確保しなければならないようなリアルタイムデータの通信に向いている。また、パケットが送信先に到達したかの返答 (ACK) を待たずにデータを送出することができるため、効率のよい通信が可能である。

今年度は、UDP パケットでの通信を導入し、アプリケーションがそれを選択できるようにした。

(3) ライブラリ化

通信サーバと VR 端末の通信を「空間共有アプリケーション構築用ライブラリ」に相応しく、アプリケーションから見て空間マネージャへの通信インタフェースとして使いやすい通信ライブラリを設計した。

(4) パフォーマンス計測

通信サーバは効率的で安全な通信経路を提供するが、そのためのトンネル化や暗号化により通信効率が落ちる可能性がある。空間共有アプリケーション構築基盤として十分な応答性を得るために、通信サーバの性能を計測する仕組みを実装し、性能を検証した。

4-2-3-1 VPN の実装

今年度は、より多くの組織のネットワーク (ファイアウォール) に対応できる汎用的な openVPN

<http://openvpn.net/>

http://www.stackasterisk.jp/tech/systemConstruction/openVpn01_01.jsp

<http://www.opensource.jp/gpl/gpl.ja.html>

を実装し、安全な通信経路を提供した。

OpenVPN は SSL-VPN を実現する代表的なアプリケーションの一つである。標準では暗号化および認証に OpenSSL を利用し、仮想ネットワークデバイス「tap/tun デバイス」を使用したレイヤ 2、レイヤ 3 レベルでの通信を可能にする。OpenVPN の主な特徴としては、通信に TCP/UDP を選択することが可能で、TCP のみでの VPN 通信しか行えない SoftEther などで起こりうる「TCP over TCP 問題」を回避することが可能である。また、Windows、Linux、Solaris、MacOS などあらゆる OS 用にパッケージが提供されているのも特徴の一つである。

VR 端末は複数のコンピュータで構成される端末 (サイト) で、通信サーバとの間にいくつもの通信路を確立し、他の VR 端末の対応するコンポーネントと通信を行う。(図 4-2-4)

昨年度の PPTP 接続の場合は、通信サーバは接続時認証を行うと共に、VR 端末のコンポ

一ネットグループに一定の IP アドレス範囲が割り当てられるようにし、それらの IP アドレスグループを「サイト」という概念で管理していた。openVPN の仕組みは、PPP 同様、通信サーバは接続時認証を行うと共に、VR 端末のコンポーネントグループに一定の IP アドレス範囲が割り当てられるようにすることができるため、昨年度開発した PPP ベースの安全な通信路上での通信プロトコルをそのまま利用することができる。

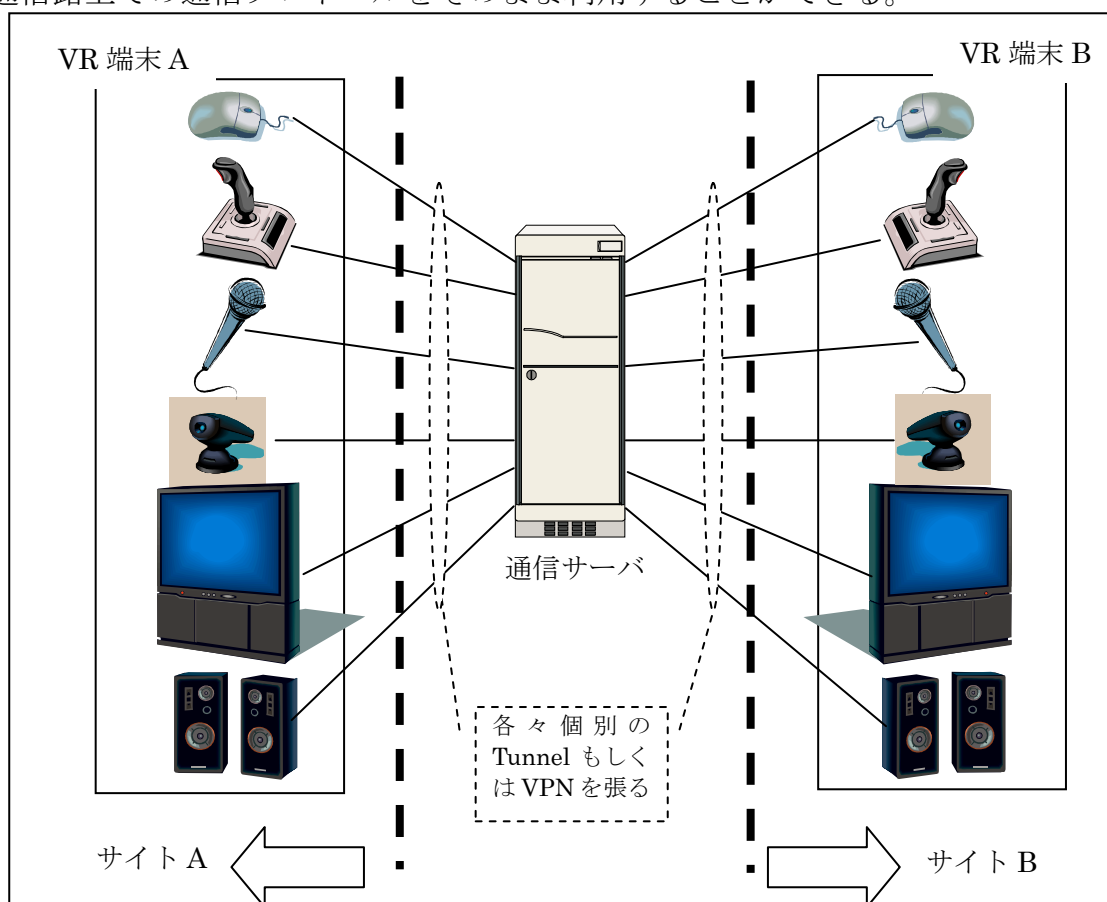


図 4-2-4 サイト概念の導入

4-2-3-2 UDP 通信の導入

昨年度は、データの確実な到達を保証し、ソフトウェア開発をしやすくすることを第一に考えたため、通信サーバを通過するパケットは TCP に限定したが、今年度は、リアルタイムデータを効率よく配信し、VR 端末の応答性や利便性を検証することを優先し、UDP 通信を導入することとした。

ビデオアバタモードにおいては、通信ポートを通信サーバと空間マネージャ間で交渉して決め、空間マネージャがアプリケーションに割り当てられたポート番号を通知する。この時、TCP 通信か UDP 通信かによって、空きポート番号を決定する主体が通信サーバか空間マネージャか変化することに注意が必要である。

アプリケーションが用いるプロトコルが TCP の場合、通信サーバがアプリケーションからのコネクションを待つため、そのポート番号を決めるのは通信サーバとなり、空間マネージャと交渉して決定する。通信サーバは受信用ポート番号で待ち受け、受信用アプリケーションはそのポートに接続してデータを受信することになる。

アプリケーションが用いるプロトコルが UDP の場合、アプリケーションがデータの到着を待つため、そのポート番号を決めるのは VR 端末側の空間マネージャとなり、通信サーバと交渉して決定する。アプリケーションが受信用ポート番号で待ち受け、通信サーバがそのポートにデータを送信することになる。(図 4-2-5)

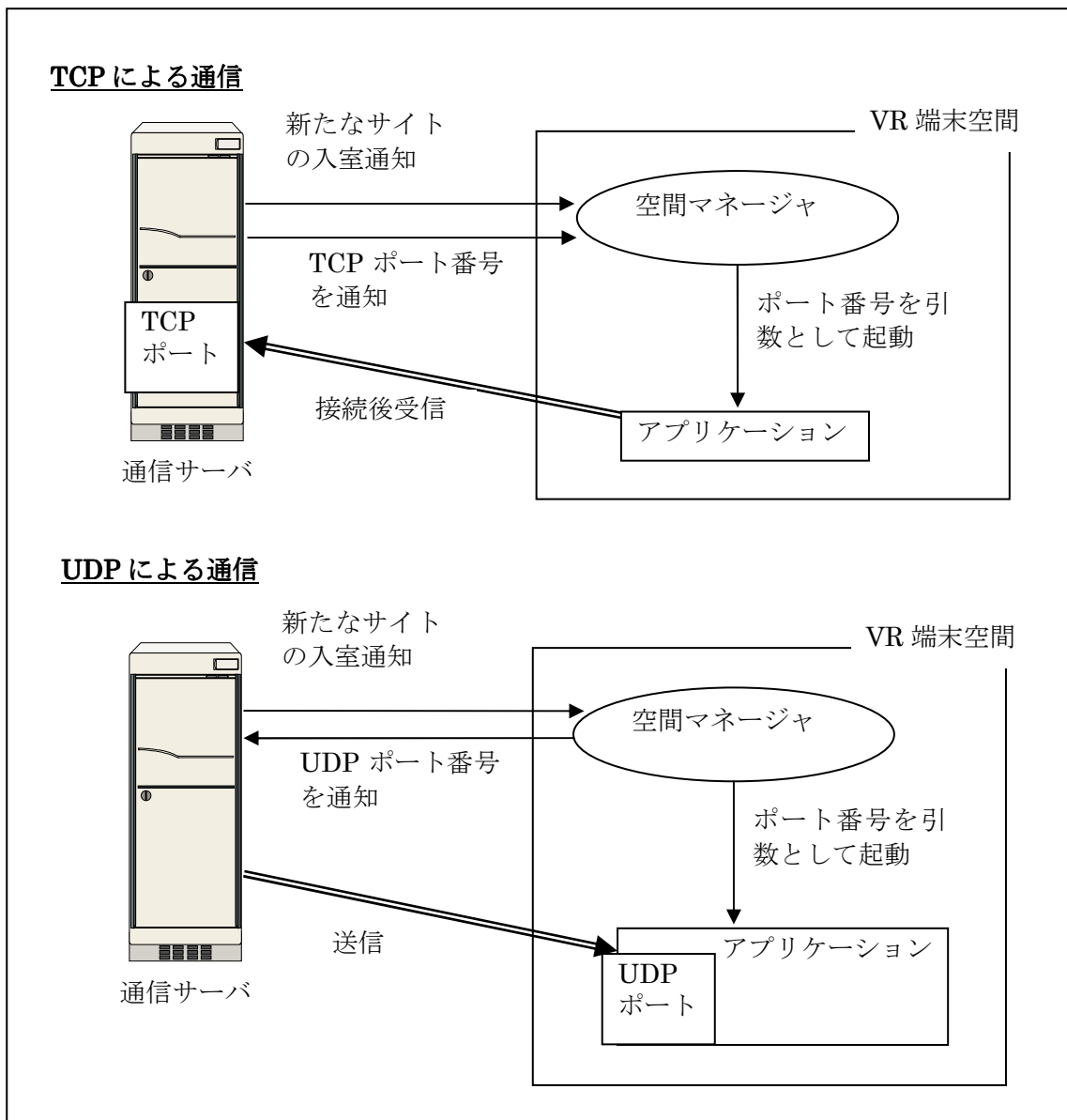


図 4-2-5 ポート番号の決定

4-2-3-3 通信ライブラリ

通信サーバとVR端末の通信を「空間共有アプリケーション構築用ライブラリ」に相応しく、使いやすいライブラリとして実装した。

「効率的な通信」の節で示したように、空間マネージャが介在し通信ポートを決定し、決定されたポート番号を使ってVR端末と通信サーバが通信を行う。VR端末上で動作するアプリケーションから見て、通信サーバは他のVR端末との通信路を提供してくれるサーバであり、その通信路の制御（状態問い合わせや新たな通信路開設等）は空間マネージャが提供してくれるサービスに見える。

アプリケーションが使いやすいライブラリは、通信制御に関するリクエストを空間マネージャに依頼し、その実装は隠蔽するような通信ライブラリとするのが妥当である。

ライブラリ化する機能としては、以下のものとした。

(1) 会議室入室

空間マネージャが通信サーバの提供する「会議室作成機能」を呼び出し、指定した会議室名が存在していなければ新たに会議室の作成を通信サーバに依頼し、その会議室に入室する。既に存在していれば、単にその会議室に入室する。「会議室作成機能」は、WEBインタフェースで呼び出すので、空間マネージャと通信サーバはhttpプロトコル

で HTML 記述を交換することになる。

(2) 会議室退室

空間マネージャが通信サーバの提供する「会議室退出機能」を呼び出し、指定した会議室からの退出を通信サーバに依頼する。通信サーバは、指定された会議室の入室者が無くなったら、会議削除機能呼び出し、その会議室を削除する。「会議室退出機能」は、WEB インタフェースで呼び出すので、空間マネージャと通信サーバは http プロトコルで HTML 記述を交換することになる。

(3) ネットワーク接続

通信サーバへの VPN 接続を行い IP アドレスの割り当てを受ける。

(4) ネットワーク切断

発行したアプリケーションの VPN を閉じる。当該サイトからの接続が全て切断されることになったら、通信サーバにそのサイトの会議室退出を通知する。

(5) 送受信チャンネル開設

アプリケーションから通信サーバにデータ送受信するチャンネルを開設する。

TCP の場合：

通信サーバから指定されたポート番号で通信サーバ側 TCP ポートに接続する。

UDP の場合：

空間マネージャが決めたポート番号でソケットオープンし、データ到着を待つ。

(6) チャンネル閉鎖

開設されていた送受信チャンネルを閉鎖する。

(7) データ送信

送信チャンネル開設によって確立された通信チャンネルを介して通信サーバにデータを送信する。

(8) データ受信

受信チャンネル開設によって確立された通信チャンネルを介して通信サーバからデータを受信する。

(9) 状態問合せ

通信サーバに会議室や通信チャンネルの状態を問い合わせる。会議室に入室している VR 端末の数や IP アドレスを知ることができる。また、通信チャンネルに繋がっている VR 端末上のアプリケーションを知ることができる。

4-2-3-4 パフォーマンス測定

空間共有アプリケーション構築基盤として十分な応答性を得るために、通信サーバの性能を計測する仕組みを実装し計測した。

性能は外部性能と内部性能に分け、通信サーバの通信ログ及び送信端末の通信ログ、受信端末の通信ログから計算し、算出した。性能計測モデルを図 4-2-6 に示す。

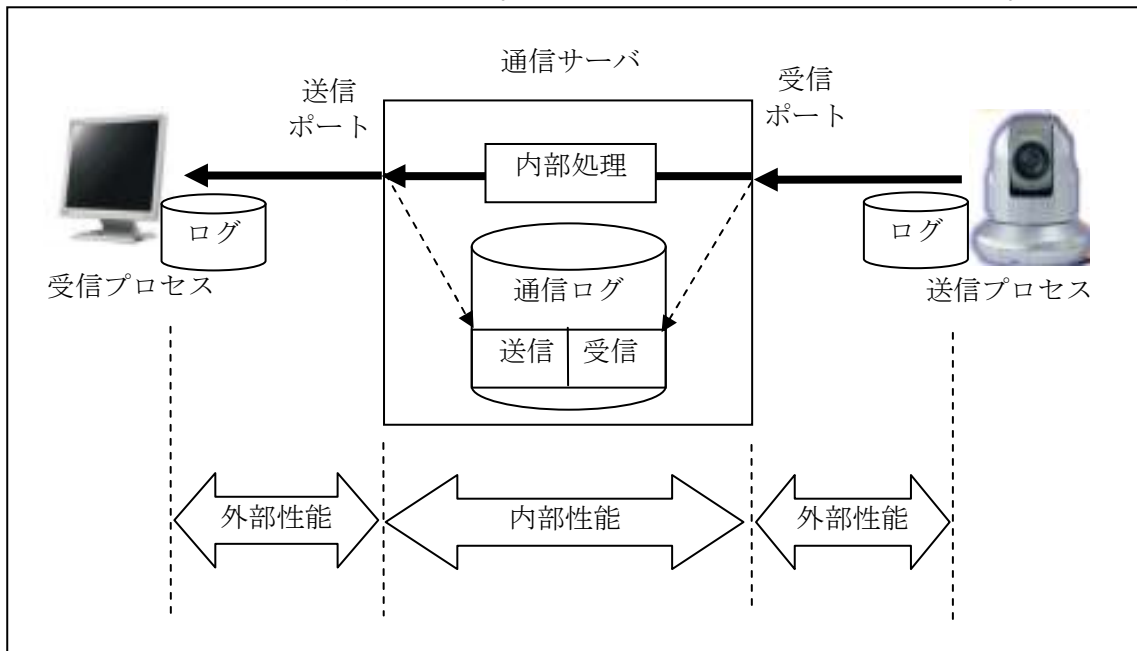


図 4-2-6 性能計測モデル

(1) WEB ページとしての性能

会議室の管理は、PostgreSQL データベースで行い、そのインターフェースは Apache で提供する WEB インタフェースとなっている。この会議室管理の性能を計測した。

PostgreSQL の性能、Apache の性能、両者の組み合わせ性能は以下のようにまとめられる。

表 4-2-3 PostgreSQL の性能

1 クライアント×100 トランザクション	10×100	100×100
528tps	1104tps	146tps

表 4-2-4 Apache の性能

1 クライアント×10000 トランザクション	10×10000	100×10000
26513kbps	47789kbps	47361kbps

表 4-2-5 動的 HTML の読み込み性能

1 クライアント×10000 トランザクション	10×10000	100×10000
364kbps	468kbps	509kbps

全体としては、PostgreSQL の性能からの制約が強く、チューニングの必要性がある。しかしながら、PostgreSQL と Apache の組み合わせで表示されるページは、設定されている会議室に関する情報確認及び会議室への入退室時にのみ参照されるもので、VR 端末同士のデータ通信にはあまり影響しない。

(2) 通信路としての性能

各通信モデルにおけるスループットをまとめると下の表のようになった。

チャットモデルに比べ、アバタモデルのスループットが極端に落ちるのは、通信サーバが受け取ったパケットをどこに転送するかを経路探索、即ち PostgreSQL の探索に時間がかかっているためだと思われる。(都度 PostgreSQL に検索をかけるのではなく) 転送先ポート番号の持ち方に工夫が必要である。また、アプリケーション 2 (VR 端末 2 台の想定) とアプリケーション 3 (VR 端末 3 台の想定) で 50%程のスループットの劣化が見られるのも、転送先ポート番号の管理方法に問題があるためだと思われる。次年度、改善していきたい。

表 4-2-6 各通信モデルの内部スループット

	チャットモデル	TCP アバタモデル	UDP アバタモデル	
			スループット	パケットロス率
アプリケーション 2×10 バイト長データ	1016kbps	905kbps	482kbps	55.5%
アプリケーション 2×100 バイト長データ	10306kbps	8644kbps	4811kbps	57.5%
アプリケーション 2× 1000 バイト長データ	102839kbps	78780kbps	47007kbps	58.4%
アプリケーション 3×10 バイト長データ	1026kbps (劣化なし)	116kbps (87%劣化)	231kbps (52%劣化)	22.9%
アプリケーション 3×100 バイト長データ	9667kbps (6%劣化)	3485kbps (60%劣化)	2251kbps (53%劣化)	22.6%
アプリケーション 3× 1000 バイト長データ	97298kbps (5%劣化)	41854kbps (47%劣化)	43566kbps (7%劣化)	65.7%

※ () 内は、同条件 (バイト長) でアプリケーション数が変わった時の劣化比率

表 4-2-7 各通信モデルの外部スループット

	チャットモデル	TCP アバタモデル	UDP アバタモデル	
			スループット	パケットロス率
アプリケーション 2×10 バイト長データ	2229kbps	1488kbps	184kbps	34.3%
アプリケーション 2×100 バイト長データ	5223kbps	6743kbps	1719kbps	0.9%
アプリケーション 2× 1000 バイト長データ	6651kbps	11641kbps	13035kbps	0.0%
アプリケーション 3×10 バイト長データ	1842kbps (17%劣化)	1493kbps (劣化なし)	537kbps (劣化なし)	73.5%
アプリケーション 3×100 バイト長データ	4033kbps (21%劣化)	6968kbps (劣化なし)	4697kbps (劣化なし)	77.4%
アプリケーション 3× 1000 バイト長データ	3080kbps (54%劣化)	8573kbps (26%劣化)	9475kbps (27%劣化)	0.0%

4-2-4 通信基盤ライブラリと既存ソフトウェアの移植性調査

既存の音声通信アプリケーションを本プロジェクトで開発した通信サーバ(CnC 通信基盤)を利用する形態へ移植し、その移植性について調べた。

4-2-4-1 機能実装

調査には東和大学で開発した音声アプリケーション：voCAL を利用した。

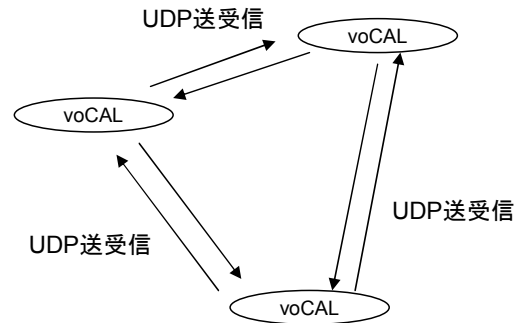


図 4-2-7 voCAL の実装

voCAL は 1 対 1 の通信を基本とし、利用者が、原則として、参加者全員の IP アドレス (+ 通信ポート番号) と、起動順序を知っていることを前提に設計されている。厳密には、先行して起動された voCAL は、後から起動される voCAL からの接続要求を受けることで通信相手を認識できるので、最初に voCAL を起動する人は、IP アドレスを知る必要は無い。しかし、少なくとも後から voCAL を起動する人は、先行して起動された voCAL の IP アドレスを引数として指定しなければならない。

3 者間通信を例に、voCAL アプリケーションを使った通信手順を説明する。

以下、最初に起動される voCAL を voCAL (1)、以下、順番に voCAL (2)、 voCAL (3) と呼ぶことにする

(1) voCAL (1) の起動

- voCAL (1) は、通信相手が存在しないので引数無しで起動され、後から起動される voCAL の接続要求を待つ TCP ソケットを生成する。引数がないので、どこにも接続要求は出さない。

(2) voCAL (2) の起動

- voCAL (2) は、voCAL (1) の IP アドレスを引数に起動され、後続の voCAL (3) からの接続要求を待つ TCP ソケットを生成する。また、引数に従い voCAL (1) からの音声受信用 UDP ポートを 1 つ生成し、voCAL (1) へ接続要求を出す。

- TCP ポートで voCAL (2) からの接続要求を受けた voCAL (1) は、voCAL (2) の音声受信用 UDP ポートに対する音声送信用 UDP ポートを生成する。(voCAL (1) ⇒ voCAL (2) の音声通信の確立) さらに、voCAL (1) は、voCAL (2) からの音声受信用 UDP ポートを生成し、TCP ソケット (voCAL (2)) に音声送信要求を返す。

- TCP ポートで voCAL (1) から音声送信要求を受けた voCAL (2) は、voCAL (1) の音声受信用 UDP ポートに対する音声送信用 UDP ポートを生成し、TCP ソケットを閉じる。(voCAL (2) ⇒ voCAL (1) の通信の確立)

(3) voCAL (3) の起動

- voCAL (3) は、voCAL (1), voCAL (2) の IP アドレスを引数に起動され、後続の voCAL (3 者では無い) からの接続要求を待つ TCP ソケットを生成する。また、引数に従い、voCAL (1), (2) からの音声受信用 UDP ポートを 2 つ生成し、voCAL (1), (2) へ接続要求を出す。

- TCP ポートで voCAL (3) から接続要求を受けた voCAL (1), (2) は、voCAL (3) が準備した音声受信用 UDP ポートに対する音声送信用 UDP ポートを、それぞれ生成する。(voCAL (1) ⇒ voCAL (3), voCAL (2) ⇒ voCAL (3) の音声通信の確立) さらに、voCAL (1), (2) は、voCAL (3) からの音声受信用 UDP ポートを追加し、TCP ソケット (voCAL (3)) に音声送信要求を返す。
- TCP ポートで voCAL (1), (2) から音声送信要求を受けた voCAL (3) は、voCAL (1), (2) の音声受信用 UDP ポートに対する音声送信用 UDP ポートを生成し、それぞれとの TCP ソケットを閉じる。(voCAL (3) ⇒ voCAL (1), (2) への音声通信確立)

NICT 通信サーバを使った実装

voCAL に CnC 通信基盤を適用することで、以下の事項を改善した。

- (1) 起動時に指定する必要がなくなる。
- (2) 起動順序を決める必要がなくなる。
- (3) 通信サーバの擬似マルチキャスト機能を利用することで、相手の数だけ送信していた音声データが、通信サーバに対する 1 送信になる。

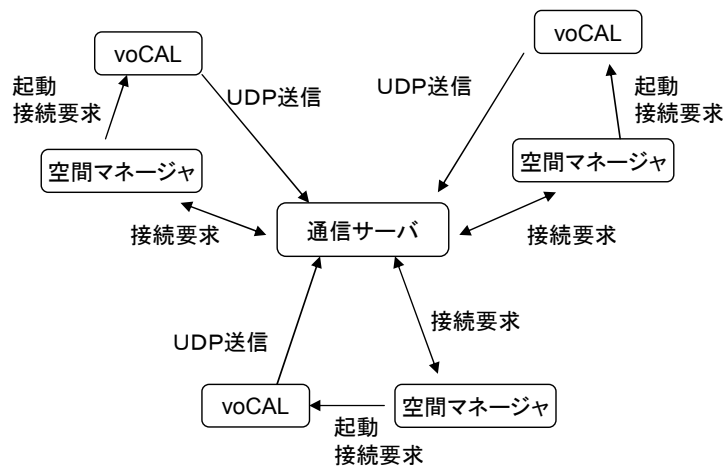


図 4-2-8 CnC 通信基盤を利用したの実装

新実装における音声データ通信の流れは次のようになる。

- (1) 事前に通信サーバに“会議名”を登録し、その会議に参加するサイトの IP アドレスを登録しておく。空間マネージャは、通信サーバの IP アドレスと会議名を登録する。
- (2) 音声データ UDP 送信開始と、空間マネージャ TCP 接続
通信基盤上では、voCAL は空間マネージャから起動される。空間マネージャは voCAL 起動時に、通信サーバのアドレスとポート番号、空間マネージャ通信用のアドレス (UDP)、ポート番号を引数として起動する。voCAL は起動するとすぐに、音声データ送信用の UDP を生成し、通信サーバへの音声通信を確立する。音声データは通信サーバによって対応する複数のアプリケーションに送信されるので、送信側アプリケーションは、1 回データを送信すれば、複数のアプリケーションにデータを伝達できる。
- (3) 音声データ受信
空間マネージャは、任意の他サイトで voCAL が起動したことを通信サーバから通知されると、自分のサイトで voCAL が起動されていないければ、起動し、起動済みであれば voCAL に音声受信準備を通知する。voCAL は、新しい音声受信用 UDP ソケットを生成し、受信待ち状態に移行する。

この結果、3者通信の場合は次の順序で接続が確立する。
voCAL(1)を利用者が空間マネージャ経由で起動する。空間マネージャは会議名を通信サーバに通知して、会議に参加者が発生したことを通知する。それを受けて、通信サーバから、各サイトの空間マネージャに通知が渡り、voCAL(2),voCAL(3)が起動され、それぞれvoCAL(1)からの受信用UDPポートを生成する。voCAL(1)は、voCAL(2),(3)の起動に従って、音声受信準備を通知され voCAL(2),(3)のための受信用UDPポートを生成する。voCAL(2),(3)も相互に音声受信準備の通知を送り合うので、それぞれの受信用UDPポートを生成する。結果、各voCALは、送信用1ポート、受信用2ポートを生成する。voCALは2つのポートから音声データを受け、それらを合成してスピーカに送る。

旧実装では、接続待ちポートは固定で、音声受信ポートはプログラム上で固定値としており、接続できる相手の数に上限があった。新実装では、ポートは外部指定なり、ポートに空きがあれば、接続相手の上限値は無くなった。

4-2-4-2 試験と評価

(1) 試験内容

実装した空間マネージャとvoCALを、東京大学、京都大学、筑波大学の3大学に設置し、CnC通信基盤を利用した音声通信試験を実施した。3拠点はJGN2ネットワークに接続している状態で、通信サーバは京都大学に配置した。

所定の設定の後、voCALを東京大学で起動すると、これに反応して、京都大学、筑波大学の音声アプリケーションが自動起動し、音声通信が行われることを確認した。

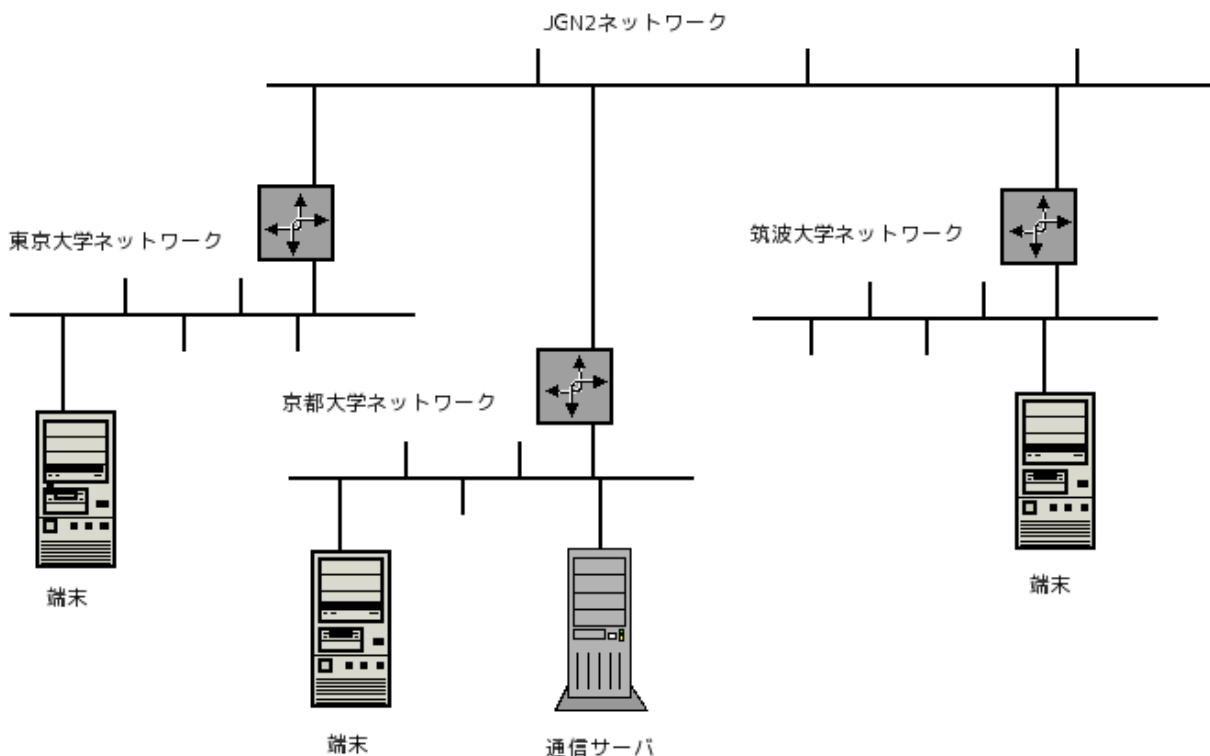


図 4-2-9 音声通信試験の環境

(2) 評価

本件は、既存通信アプリケーションを本プロジェクトの成果物である通信基盤に移行することの容易性と、通信基盤を利用することの優位性についての評価を目標とした。今年度、通信サーバに UDP 通信機能が盛り込まれたことによって、音声アプリケーションのような UDP 通信が不可欠なアプリケーションも、CnC 通信基盤への移行が可能になった。CnC 通信基盤を用いた voCAL プログラムは、通信形態の変更により、プログラム構造が簡素になった。いくつかのデータ構造の変更が必要であったが、CnC 通信基盤の利用により、通信相手数の制限の排除や自動応答などの機能を組み込むことができるようになった。移行の手間を考えてもメリットが多いのではないかと考える。

また、通信サーバの UDP 通信も、3 者間通信では通信データの欠損も少なく、十分に実用に耐えうることを確認した。

4-2-5 まとめ

空間共有ライブラリの開発は、遠隔授業実証試験と平行となったため、実用的な機能実装は先行したが、公開用のライブラリ整備が遅れた。平成 19 年度はライブラリ整備とドキュメント整備を実施し、利用に供するものに仕上げる。

通信基盤システムは、予定通りの実装がほぼ終了し、本年度に実施した API 設計に従って、平成 19 年度は実装を行う予定である。音声通信アプリケーションの移植と利用も予定通り実施できた。しかし、まだ、アドバイザー・グループのプロトタイプや実証試験への利用が無い場合、基本性能測定はできているが実際の利用に対しての利便性が未知数である。平成 19 年度は、さらに、既存ソフトウェアを通信基盤システム対応にする作業を行うことで、既存システムとの親和性を高めていく予定である。

4-3 知識創造プロセスの支援のためのデータベース開発

4-3-1 序論

本サブテーマでは、IPT 環境内からのデータベース利用のための入出力インターフェースの研究開発を、実用的なプロトタイプシステムを実装しながら進めてきた。

平成 17 年度は、KJ 法を IPT システム上に構築し、その中で PDA (Personal Digital Assistant) を用いたユーザインタフェースを開発した。その結果、KJ 法は没入型の立体視システムよりも、タイル型ディスプレイの方が高解像度で安価なシステムを構築できることから、本システムの実装をタイル型ディスプレイ向けに実装することとした。そこで、本年度は、柔軟性の高いタブレット PC を使って、タイル型ディスプレイ向けのユーザインタフェースを開発することとした。

また、一般にコミュニケーションは音声だけでなく表情やボディランゲージが併用される。ここでは、タイル型ディスプレイの中心にカメラを配置することで、相互に相手の表情を見ながら会話が可能となるシステムを開発した。

4-3-2 タイル型ディスプレイ向け KJ 法システムの開発

タイル型ディスプレイは格子状に配置した LCD と各 LCD に視覚情報を提示するディスプレイノード、アプリケーションの実行とディスプレイノードへ提示する画像情報をコントロールするマスターノードから構成される。

大画面表示は、プロジェクタを使うことでも可能だが、数メガピクセルの視覚情報を拡大して投影するために画素が粗くなる。一方、タイル型ディスプレイでは数メガピクセルの解像度の LCD を格子状に配置することで、大画面で、且つ、LCD の数に比例して高精細な視覚情報を提示するディスプレイの構成が可能となる (図 4-3-1)。

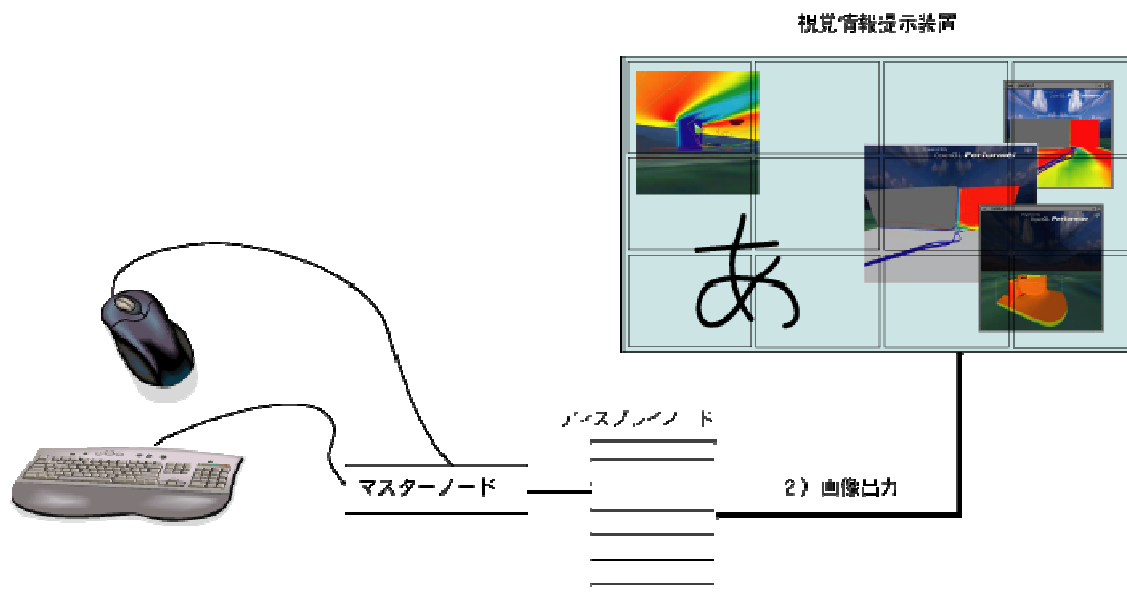


図 4-3-1 一般的なタイル型ディスプレイの構成

タイル型ディスプレイは、高解像度の画像や動画を同時に多数提示し、複数人による協調作業や高解像度画像の参照に用いることができる。この際、ユーザはタイル型ディスプレイの前に立って利用する場合が多いので、キーボード・マウスといった従来のユーザインタフェースの利用が困難になる。さらに、複数のユーザがタイル型ディスプレイに提示されている視覚情報を操作するためには、従来のような一つの入力装置では効率的な協調作業を行えない。そこで、タイル型ディスプレイに提示される複数の画像の操作を行うた

めに、タブレット PC を活用したユーザインタフェース(以下 KJ++ui)を開発した (図 4-3-2)。

KJ++ui にはタイル型ディスプレイに提示されている視覚情報と同じ情報が尺度調整 (主として縮小) されて提示される。KJ++ui 及びマスターノードはデータベースに格納している画像を表示する。KJ++ui とマスターノードは無線で接続する。視覚情報はマスターノードで生成された情報を KJ++ui に転送するのではなく、KJ++ui とマスターノードそれぞれ個別に生成する。

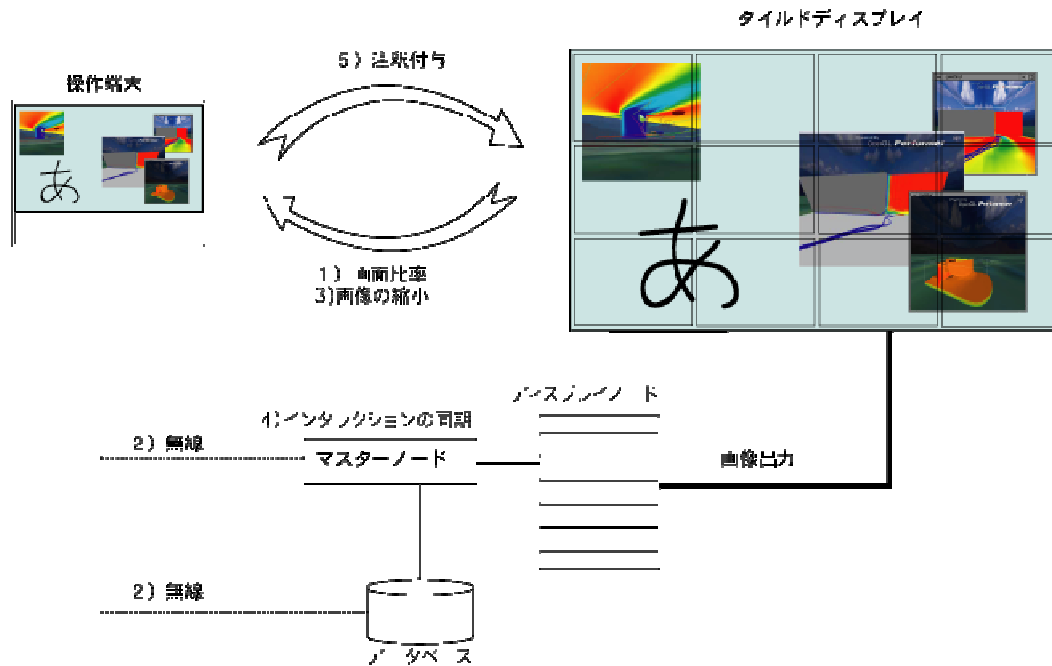


図 4-3-2 KJ++ui のシステム構成

(1) 初期化 (解像度、縦横比情報の取得)

タイル型ディスプレイは、構成される LCD の数によって縦横比が変わる。マスターノードはタイル型ディスプレイの環境設定を KJ++ui に送信し、タイル型ディスプレイと同じ縦横比のウィンドウが KJ++ui に表示される。このウィンドウはタイル型ディスプレイを表し、ウィンドウの大きさとタイル型ディスプレイの縮尺値が設定される。

(2) 画像の登録

本システムで扱う全ての画像は事前にデータベースに格納しておく。

画像登録には Web ブラウザを用いる。Web サーバはデータベースと接続されており、アップロードされた画像をデータベースに格納する。画像のアップロード時にコメント、登録者、検索時に用いるパラメータと値を付加情報として登録する。

(3) 画像の表示

画像の取り出しは、KJ++ui を用いる。

ユーザは、画像データベースにアクセスして画像一覧を KJ++ui に表示でき、そこから任意の画像を選択し KJ++ui のウィンドウに表示できる。このとき、アップロード時に登録したパラメータと値を用いて絞り込み検索も可能である。

KJ++ui で選択された画像の属性データは、KJ++ui からマスターノードに送信される。マスターノードはその属性データを用いてデータベースを参照し、タイル型ディスプレイに画像を表示する。

その結果、操作端末とタイル型ディスプレイには同じ画像が表示されるが、操作端末は

表示領域が限られているので、初期化時に設定された縮尺値を基に縮小されて画像は表示されており、タイル型ディスプレイにはデータベースにフル解像度の画像が表示される。

(4) 画像の移動操作

画像の位置はタイル型ディスプレイと KJ++ui の対応した位置に表示される。タイル型ディスプレイに表示されている画像の位置は、KJ++ui に表示されている画像をカーソルキーもしくはスタイラスペンで変更できる。操作端末の画像座標がマスターノードに転送されタイル型ディスプレイに表示されている画像の位置も変わる。このとき転送される画像の位置座標は KJ++ui とタイル型ディスプレイの比率に合わせて変換される。

(5) 注釈の挿入

本システムでは、タイル型ディスプレイに表示されている画像に対して、KJ++ui を通して手書きで文字や図形などの注釈付与ができる。KJ++ui 上で手書き描画される注釈はリアルタイムにマスターノードへ転送され、タイル型ディスプレイ上に反映される。タイル型ディスプレイに描画される注釈の大きさは KJ++ui と提示装置の画面の比率に応じて拡大される。

(6) 複数端末におけるコラボレーション

1 台のタイル型ディスプレイに対して、複数の KJ++ui が接続可能である。一つの KJ++ui がマスターノードに転送したユーザインタラクションは他の KJ++ui にも転送され、全ての KJ++ui とタイル型ディスプレイに表示されている情報が同期する。また、注釈もタイル型ディスプレイのみではなく、他の KJ++ui にも転送され、同じ注釈を共有する。

KJ++ui とタイル型ディスプレイに表示された画像情報、位置情報などは全て保存が可能で、継続作業も可能である。保存される情報は、各 KJ++ui に保存される。

KJ 法システムのシステム構成とクイックスタートガイドを別紙 3 に示す。

4-3-3 タイル型ディスプレイを用いた遠隔コミュニケーションにおけるアイコンタクトの評価実験

複数のユーザ間で PC を用いた遠隔コミュニケーションとして、数多くのアプリケーションが提供されているが、PC 画面に対して映像の表示サイズが小さい。サイズを拡大しても映像の解像度が低く不鮮明のため、相手の存在感の提示に不十分である。一方、高品質なビデオ会議システム製品の普及が進み、様々な現場で利用されている。これらシステムの映像の大画面提示にプロジェクタの活用が考えられる。しかしプロジェクタの表示解像度の仕様によって、スクリーンサイズを大きくしても、大画面上に粗く不鮮明な画像を提示することになる。

一方、これらのアプリケーションや会議システムで使用する映像撮影用のカメラは、表示ディスプレイの上や横にもしくは三脚を用いてディスプレイとずらして設置することが多い。このような設置方法では、カメラを参加者に向けることは可能であるが、参加者の視線はディスプレイに表示される遠隔地側の参加者に向けられる。そのため、お互いにアイコンタクトによるコミュニケーションが困難である。

これらの問題に対して、複数の液晶ディスプレイをタイル状に配置し、大画面かつ高解像度表示が可能なタイル型ディスプレイを用いて、広域ネットワークで介した 2 拠点にそれぞれ設置した遠隔コミュニケーション環境を構築した。さらにディスプレイの中央に撮影用に小型の IEEE1394 カメラを設置することで、画面上の相手とのアイコンタクトがとれる遠隔コミュニケーションの可能性について実験的検討を行った。

本研究ではタイル型ディスプレイに対して複数のグラフィックス・ソースを仮想的な高解像度フレームバッファを用いてピクセル情報をストリーミング配信するために、SAGE (Scalable Adaptive Graphics Environment) を適用した。SAGE はイリノイ大学の EVL で開発された高解像度表示による協調可視化環境を支援するためのグラフィックストリーミング用アーキテクチャである。SAGE を用いることによってスケーラブルなタイル型ディスプレイの構築と高解像度の画像を遠隔地で高速に表示することが可能となる。

構築したタイル型ディスプレイシステムは、4 台の表示ノードと 8 台の液晶ディスプレイ、および 1 台の管理ノードで構成し、液晶ディスプレイを 4x2 の配置で組み合わせている(図 4-3-3)。各ノードの OS は SUSE Linux(kernel 2.6) で、CPU は Pentium4 3GHz、メモリは 1GB である。グラフィックスボードとして、2 画面出力に対応した nVidia Quadro FX1400 を用いた。各ディスプレイのサイズは 17 インチで、解像度は 1280x1024 ピクセルに設定している。管理ノードと全表示ノード間はギガビットイーサネット(1Gbps) で接続されている。IEEE1394 カメラ(Point Grey Research 社製 Flea) は図 4-3-3 に示すようにディスプレイの中央に設置することで、参加者を正面から撮影できるようにし、アイコンタクトを取りやすい環境になると考える。カメラ画像のキャプチャ・アプリケーションは IEEE1394 カメラで取り込んだピクセル情報を glDrawPixels を用いて描画する。キャプチャ・アプリケーションは SAGE Application Interface Library (SAIL) の API コードをアプリケーションのソースコードに組み込むことで、SAGE 上で実行可能なアプリケーションとなる。

ネットワーク構成を図 4-3-4 に示す。本研究では東和大学(福岡市)と京都大学の 2 拠点で構成し、拠点間は SuperSINET 上を IPSec による VPN で接続している。回線の帯域として、京都大学は 1Gbps、東和大学は 10Mbps で SuperSINET と相互接続されている。それぞれのタイル型ディスプレイの中心に設置されている IEEE1394 カメラで撮影された映像をリモートに転送し、リモートで撮影された映像をローカル側のタイル型ディスプレイに表示することにより、対面型の遠隔コミュニケーションを実現させる。

本環境において、ディスプレイの中央位置にある IEEE1394 カメラをディスプレイの上端及び下端にずらして設置位置を変化させた場合において、撮影された人物の視線方向がどのように遠隔地の画面に映っているかについて評価実験を行った(図 4-3-5)。なお、撮影された人物は視線方向をディスプレイの中央を向くようにする。その結果、カメラを上部

に設置した場合は視線が下向きに，下部の場合は上向きになる傾向が見られた．それに対して，カメラが中央位置にある場合は視線方向が正面を向いていて，相手の視線を感じる傾向が高くなることがわかった．この結果より，本環境での遠隔コミュニケーションにおいて，タイル型ディスプレイ上の相手とアイコンタクトを実現する可能性を示すことができた．

今後はカメラの位置と視線方向の関連性について，様々な観点から定量的な評価実験を進めていき，詳細な検証を実施する予定である．

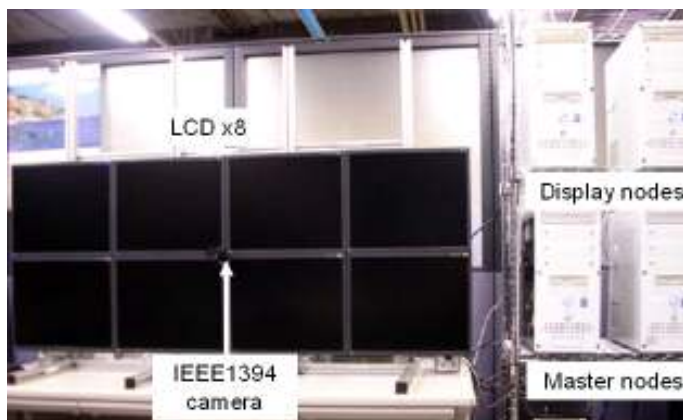


図 4-3-3 タイル型ディスプレイの構成



図 4-3-4 ネットワーク構成



図 4-3-5 アイコンタクトの評価実験

4-3-4 まとめ

本サブテーマでは，KJ 法アプリケーションを開発しながら，インタフェース機能を開発してきた。本年度はタブレット PC を活用したタイル状ディスプレイ用のインタフェースシステムを開発した。

また，遠隔コミュニケーションにおけるアイコンタクトの実現に向けた試験システムの開発を実施した。

平成 19 年度は，実証試験と評価，それに伴う改良試験を実施する予定である。

4-4 アドバイザリ・グループの需要研究.

4-4-1 序論

本研究開発の目標は、先端技術の開発と同時に、その技術を用いた収益事業の構築である。したがって、技術シーズと市場ニーズをマッチングさせる活動が重要である。そのため、平成16年度から、本研究の主旨に賛同いただいた民間企業、大学研究所をアドバイザリ・グループとして参加いただき、ニーズのヒアリングを行い、本技術を用いたアプリケーション構築のプロトタイプ選定を行ってきた。

本年度は、その中から遠隔授業のプロトタイプ開発と実証試験、会社の本社・支店間のTV会議接続試験、建築分野向け研究成果PRシステムのプロトタイプ開発を実施した。

同時に新規プロトタイプ開発候補として、タイル型ディスプレイを高等教育に利用する“E-黒板(通称)”の市場調査を実施した。

また、オープンソースとして提供予定のCnC基盤ライブラリの普及推進母体としてN3VR研究会との研究活動を継続的に行った。

4-4-2 遠隔授業と研究開発者のコラボレーション

テレ・イマージブ・カンファレンスの応用アプリケーションとして遠隔授業があげられる。従来のビデオ会議をベースにした遠隔授業システムでは、講義を中心とした授業には適用できるが、体験実習やインタラクティブなコラボレーションが要求される授業には利用できない。ここでは空間共有アプリケーション構築用ライブラリを用いた体験型の遠隔授業システムのプロトタイプを開発し、筑波大学の実験の実習授業に使用した。

プロトタイプで利用できる機能としては、ビデオアバタによるコミュニケーション機能、データベースを用いたPowerPoint教材や3次元CGモデルの共有機能、指し棒によるポインティング機能、仮想キーボードを用いたホワイトボード機能等である。また利用した授業は「没入型VRプログラミング」の実験演習の授業で、9人の学生を対象にした約40分間の講義に利用した。実験環境としては、JGNIIつくばリサーチセンターの没入型拡張現実感ディスプレイAR Viewと3面スクリーン構成のCS Galleryの間をつくばWANで接続したネットワーク環境を使用し、教師はAR View側に、学生はCS Gallery側に集まった。下図4-4-1は遠隔授業の実験の様子を示したものである。

実験では授業を行った後アンケートにより学生の感想を聞いた。その結果、ビデオアバタの映像品質についてはやや不満があったが、ビデオアバタを用いた教師との質疑応答のコミュニケーションや指し棒による指示は円滑に行うことができ、概ね対面授業と同様に講義に集中することができたとの感想が得られた。

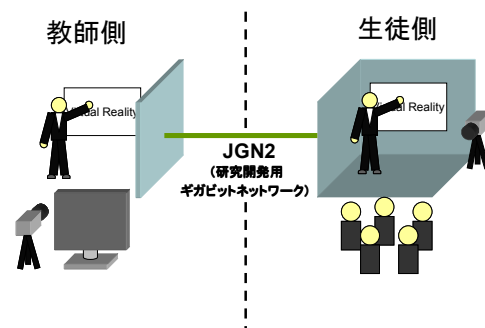


図 4-4-1 空間共有機能を用いた遠隔授業実験の様子

4-4-3 マルチ画面システムを使った空間共有システムの開発と実装実験

マルチ画面システムを使った、ネットワーク会議システムアプリケーション (FreshVoice) による遠隔会議の実用性についての検討をおこなった。

(1) 環境

ネットワーク会議システムでは、一般的に端末を含めハードウェアが固定され、各会議サイトでも専用のハードウェアが必要になる。FreshVoice では、端末側は特別なハードウェアを必要とせず、ヘッドセットと USB カメラが接続されている WindowsPC であれば利用することができるなどの特徴をもつ。本件では、会議サーバを KGT 社内に設置し、東京本社-大阪支店間の社内 LAN (1Mbps) 上で試験を実施した (図 4-4-2)。端末に使ったマシンは双方、出力を2つもつグラフィックスボードを2枚挿し、4画面出力としたシステムである。

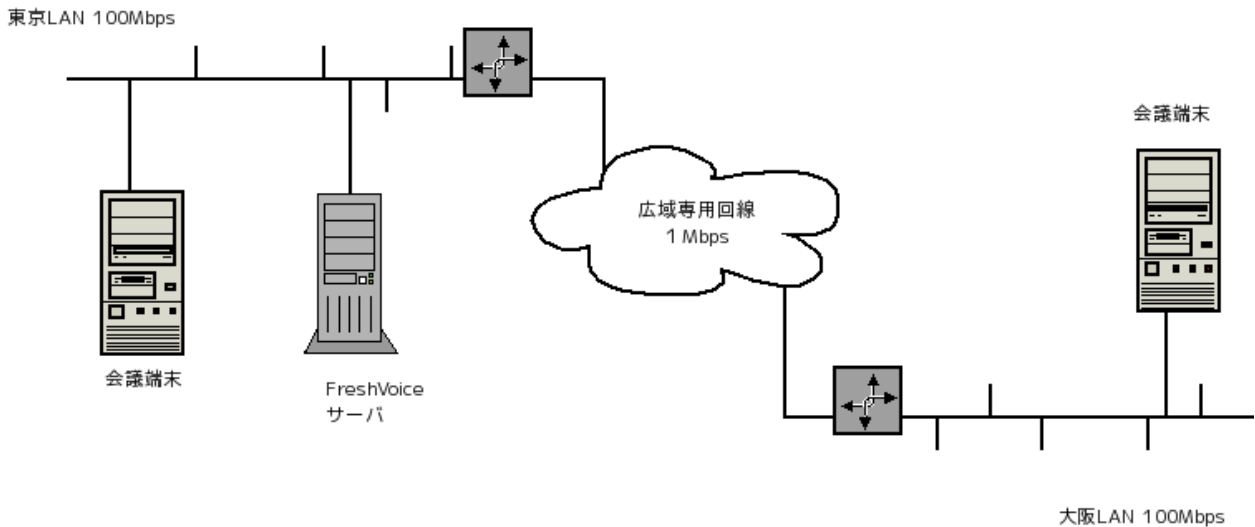


図 4-4-2 東京-大阪接続試験環境

(2) 試験内容

FreshVoice では、画像データの通信時に品質の設定ができる。試験ではこれらの設定を変更しながら通信トラフィックの変化を計測し、実用性について検討した。試験した設定を表 4-4-1 に示す。

表 4-4-1 試験パラメーター一覧

実施時間帯	クオリティ	フレームレート	解像度
13:00 - 14:00	Medium Quality	6	320x240
14:00 - 15:00	Low Quality	3	160x120
15:00 - 16:00	Low Quality	3	320x240
16:00 - 17:00	Low Quality	6	160x120
17:00 - 18:00	Medium Quality	3	160x120
18:00 - 19:00	High Quality	6	320x240

クオリティは、画像劣化度を示すパラメータ。

図 4-4-3、4-4-4 は、上記設定条件を 13:00 から 1 時間ごとに変更していったときの端末のトラフィック状況である。グラフ横軸の数値が時刻を示している。

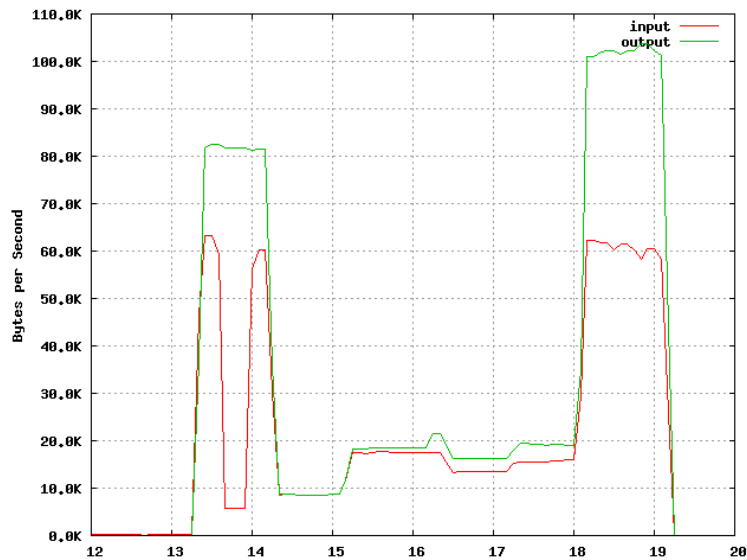


図 4-4-3 東京側端末のネットワークトラフィック状況

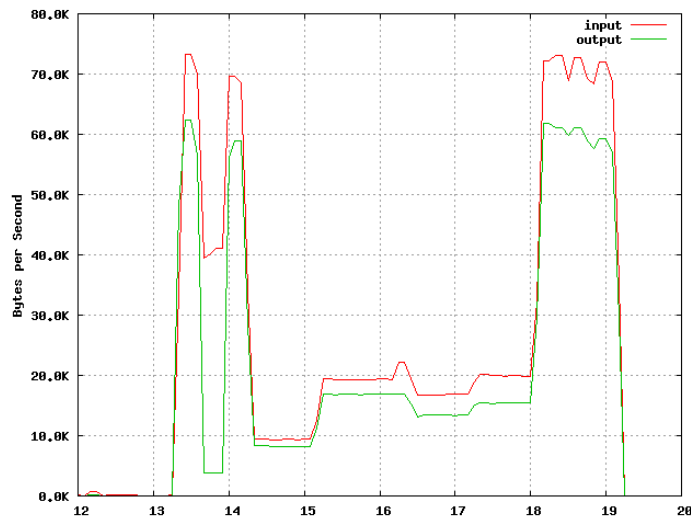


図 4-4-4 大阪側端末のネットワークトラフィック状況

13:00-14:00 の間に深い谷があるのは、大阪側端末でアプリケーションが異常終了して再起動したためである。

試験は、通常業務中に実施した。通常業務に支障のない範囲で試験する必要があったので、あまりに高負荷となるパターンは試験できなかったが、クオリティ、フレームレート、解像度を高くすれば、ネットワーク負荷も高くなることわかる。本試験の範囲では、18:00 - 19:00 の設定(High Quality, 6 フレーム/秒、解像度 320x240)が最大負荷となった。この状況では、ネットワーク全体に大きな影響があった。例えば、大阪側でネットワークを利用する他アプリケーションが、異常動作や、異常終了する現象が発生した。

14:00 - 18:00 までのパラメータ変更による負荷の変化をみると、フレームレートは、クオリティ、解像度より影響が少ない。

(3) 常時接続

本システム試験の目的は、マルチ画面を窓と見立て、双方向に音声、カメラ画像を送受信することによって、遠隔地間の空間を窓で隣接する空間となるような状況を実現するものである。

オフィス全体が見渡せる位置にカメラを設置し、マイクもカメラと同一位置に設置。スピーカはディスプレイ付近に設置した。図 4-4-5 に大阪オフィスを撮影している様子を示す。



図 4-4-5 常時接続試験の様子（大阪オフィスの映像）

数日間、常時接続試験を行ったところ、次のような感想が得られた。

- (a) 画像解像度が低いため、マルチディスプレイで大画面にしても、画像が粗く、近距離では相手の認識が困難であった。5 m程度離れた場所からは、漠然と先方の様子を見ることができて良かった。（誰も居ないとか、誰かが帰社した、などが判る）
- (b) カメラ位置と、ディスプレイ位置が一致していないため、窓というよりは監視カメラのような映像になってしまった。そのため、見られている側では、常に監視されているように感じさせた。
- (c) スピーカ近くに席がある人は、スピーカからの雑音が気になった。

(4) まとめ

マルチ画面環境を使って、市販のネットワーク会議システムの運用を試みた。

1台のPCに2枚のグラフィックスボードを挿入し、グラフィックスボード1枚が2画面出力を行う構成の4画面構成のシステムと FreshVoice（会議ソフトウェア）を用いた。その結果、FreshVoiceでは、2画面出力（グラフィックスボード1枚）は可能であったが、複数のグラフィックスボードを跨ぐ表示はできなかった。また、2画面に出力した際、縦横の画面比率が考慮されず、横に伸びて表示された。現在の市販会議システムは、マルチ画面が考慮されていない場合があることがわかった。

東京・大阪を社内LAN(1Mbps)で接続する試験では、解像度 320x240、6フレーム/秒（High Quality）という条件で、ネットワーク帯域の50%を占有してしまい、他の作業に影響を与えてしまった。この程度の環境では、マルチ画面が有効となるほどの解像度での映像転送は不可能であることがわかった。

常時接続試験では、カメラの位置、マイク位置、スピーカ位置を工夫して、より窓のような状況を作り出す必要があることがわかった。

4-4-4 高等教育支援システムの市場機会の検討

本プロジェクトでは、基盤ソフトの応用用途として、高等教育支援システム(以下「E-黒板」と呼ぶ)の市場機会を検討している。検討概要は、以下の通りである。

(1) 市場の状況

「E-黒板」と呼ばれる市場が顕在化している背景には、小・中・高の教室(全国40万教室)をネットワーク化し、教室にプロジェクタや電子情報ボードを導入して、「理解しやすい授業」を目指す文科省の教育施策がある。米国イリノイ大学などでは、タイル状に接続した多画面に、学生の PC 画面を電子黒板に、一方方向であるが、表示している事例があり、今後、この流れは、大学などの高等教育にも進むものと予想される。

(2) 需要調査

今年度、京都大学(再委託研究先)で以下の視点から有効性を調査する準備を開始した。

- ・3D 画像を含めた画像表示を用いて、教師と学生の間で「双方向コミュニケーション」を行う。
- ・さらに、遠隔地の教室とネットワークを越えたコラボレーションを行う。(図 4-4-6)

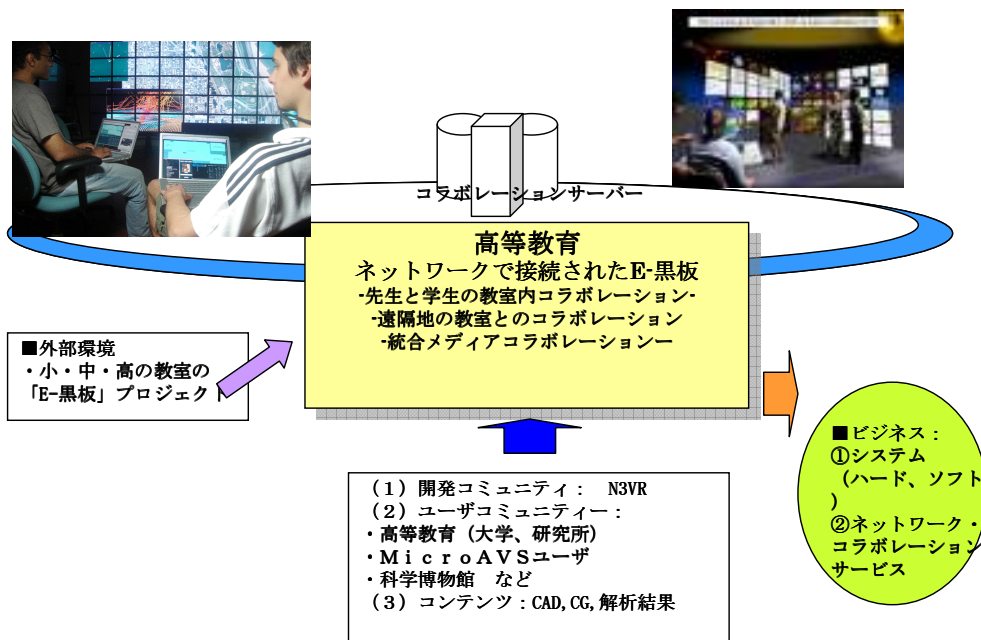


図 4-4-6 高等教育における遠隔教育ニーズ

(3) 「E-黒板」用アプリケーション:KVS

「E-黒板」用に有効なアプリケーション・ソフトを企画する必要があると考え、京都大学で開発し配布している無償ソフト「KVS」(Kyoto Visualization System) を調査した。この調査の結果、「E-黒板」用に有効なアプリケーション・ソフトであると判断されたので、来年度、CnCライブラリで移植を行う計画である。

KVS 概要

スペック :

- ・ 可視化アルゴリズムや数値計算アルゴリズムを実装した C++クラスライブラリ
- ・ 可視化ユーザ向けというよりは、可視化開発者向け
- ・ KVS を自分のプログラムにリンクし、可視化プログラムを書く
- ・ Qt (キョウト) をサポート (GUI アプリも比較的簡単に開発可能)
- ・ ソースコード公開

機能：

- 基本的な可視化アルゴリズムを提供。
- IEEE Vis を中心とした最新アルゴリズムの実装。
- 非構造データへの対応。
- AVS、VTK、OpenDX と同様に可視化パイプラインの構成。
- シーングラフも構成可能。
- マルチスレッドなどの機能も簡単に利用可能。

利用者の現状

大学を中心とした研究者が利用している。

京都大学内メディア専修コース、可視化情報学会の OpenGL 講習会、等で普及促進活動を実施中である。

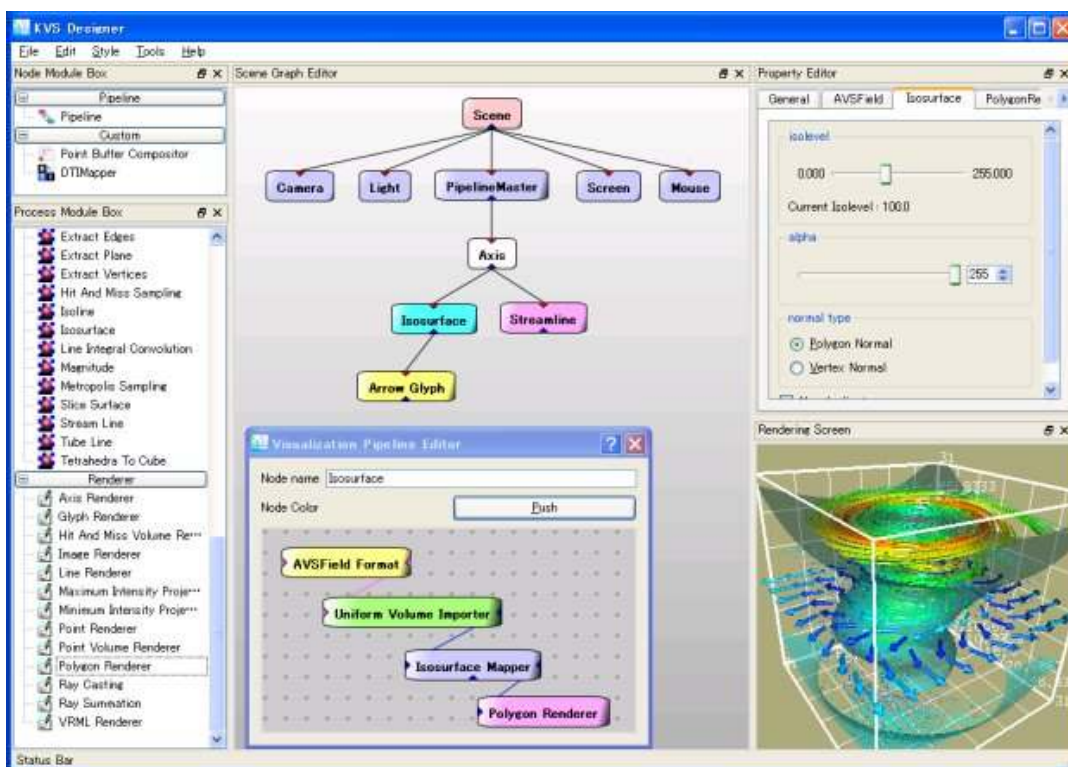


図 4-4-7 KVS を用いたアプリケーション動作画面

(参考文献) 京都大学センター広報

<http://www.viz.media.kyoto-u.ac.jp/kvs/archives/KVS-ACCMS-newsletter.pdf>

4-4-5 アドバイザリ・グループとの連携によるプロトタイプ開発

今年度のアドバイザリ・グループの需要研究で、建築関係の会社と研究成果 PR 用プロトタイプ版の開発に至った。先方のニーズを列挙すると以下となる。

- (1) R&D の成果（数値シミュレーション結果）をダイレクトに製品（建築データ）と重ね合わせることで大幅な工数削減になる。
- (2) デスクトップで表示できるとともに、そのまま大画面へも表示できることは展示会では有効である。
- (3) 同様の PR は大規模建築の際、役所や住民への説明で有効と考えられている。
- (4) 今後の展開としては、ネットワーク化のニーズがある。各地の住宅展示場には様々なコンセプトの住宅を建てていて、それぞれの会場に専門性の高いアドバイザが居る。これらの会場、および、研究所をデスクトップタイプで接続すれば、来場者に各種住宅展示場の住宅を見せ、専門性の高い説明が可能となる。VRタイプを使って、すべての住宅展示場を仮想的に接続することには夢があるが、生涯に何度も無い住宅購入決定の動機付けができる程度の品質が望まれる。

これらの要望の第一ステップとして、図 4-1-11 に掲載した建築 CAD と室内気流解析の OpenGL フュージョンによる合成表示システムを開発した。その成果は、「ふるさと建材・家具見本市 (H18. 11. 15-17: 東京ビッグサイト)」で展示された(図 4-4-8)。



図 4-4-8 「ふるさと建材・家具見本市」におけるフュージョンプロトタイプ展示

また、自動車関連会社では、プロトタイプ作成までは実施していないが、設計部門での設計者と管理者を結ぶコラボレーションというニーズが出てきている。

- (1) 設計部門の管理者は、設計途中の経過をモニタリングしたい。
従来は図面に描いていたので、現場を回覧すれば設計経過が一瞥できた。
しかし、現在は小さなモニターに向かって作業をしているので、それができない。
設計経過の様子を大画面に並べて一瞥できれば設計者間の矛盾を発見できるかもしれない。また、そこから新しい創造できるかもしれない。
- (2) 車体設計者、エンジン設計者、マーケティング間でのコラボレーションもある。

4-4-6 N3VR 研究会における CnC 基盤ライブラリの啓蒙活動

N3VR 研究会主催のテレ-immージョン技術研究発表会が 3 回実施された。

N3VR 研究会ホームページ (<http://www.n3vr.org/>)

平成 18 年 5 月 12 日 第 1 回研究会 筑波大学公開講義室

平成 18 年 10 月 10 日 第 2 回研究会 京都大学学術情報メディアセンター 北館 3F

平成 19 年 1 月 26 日 第 3 回研究会 岩手県立大学アイーナキャンパス

この中で継続的に本研究の成果を公表し続け、CnC 基盤ライブラリの利用を呼びかけてきた。(参照：5-1 研究発表・講演等一覧 #18-10,21,22,27,29)

第 2 回研究会では、VR のメッカである米国イリノイ大学から Jason Lee 氏を招待して、最新の世界での研究動向について講演をいただき、以下の事項が判った。

- (1) 北米では CAVE 型よりもタイル型ディスプレイの研究が増えてきている。
- (2) タイル型ディスプレイを教育に利用する事例が紹介された。
- (3) 高解像度の立体タイル型ディスプレイも登場した (図 4-4-9)。

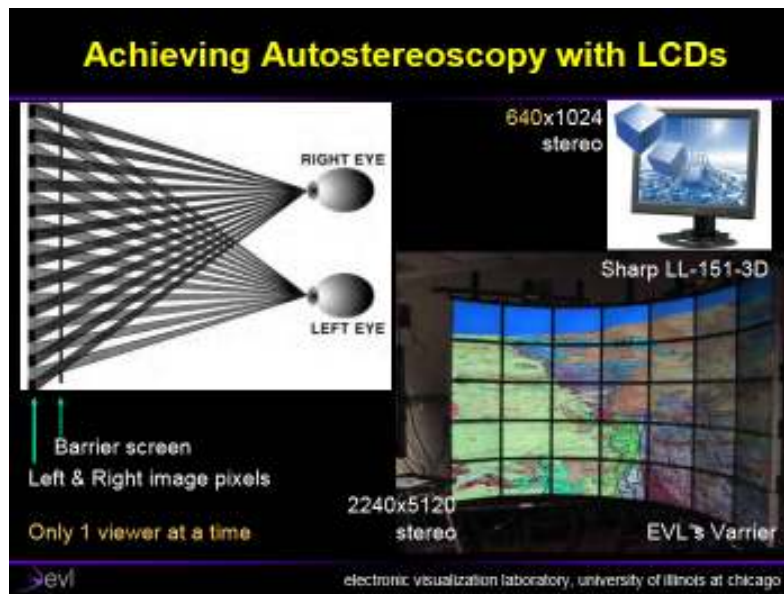


図 4-4-9 LCD を使った高解像度タイル型立体視ディスプレイ

4-4-7 まとめ

遠隔授業の実証試験により、IPT システムを利用した遠隔授業は、従来の TV 会議システムによる遠隔講義と比較して、学生に対面授業に近い感覚を与えることができると思われる。しかし、品質面では、まだまだ改善の余地は残されていることが解った。

今後は、タイル型ディスプレイシステムでも遠隔授業を試み、通称“E-黒板”の事業の実現可能性試験を実施する予定である。

本社・支社間 TV 会議システム接続試験は、テスト環境 (回線速度：1Mbps) では、タイル型ディスプレイの解像度を有効利用できないことが判った。また、単に映像と音声を共有するだけでは、気配などが無いため、ワークスペースに配置したのでは監視されているという印象を払拭できないこともわかった。

建築分野向けの研究成果 PR システムのプロトタイプ開発では OpenGL フュージョン技術が有効に活用できた。本年度は秋の展示会に向けての開発という手ごころな目標を共有できたが、次年度の予定は未定である。交渉によって、本研究開発の時間スケールで有効な共同作業を提案していく予定である。また、今後も、医用分野など、その他のプロトタイプ開発の機会も並行して調査を継続する。

N3VR 研究会との連携は継続する。

4-5 総括

平成18年度は、研究成果の社会還元、事業化の第一歩として、公開用の整備や、実証実験に必要な利用者ニーズを満たす機能拡張を各サブテーマで実施した。

IPT用VR基盤ライブラリのOpenCABIN libraryは無償公開に向けた整備を実施した。

同サブテーマのOpenGLフュージョン技術は商用化が実施され、フュージョンVRとして製品化された。この製品は財団法人りそな中小企業振興財団主催（日刊工業新聞共催）の第19回中小企業優秀新技術・新製品賞のソフトウェア部門の優秀賞を2007年3月29日に受賞した。また、プロジェクト内でのプロトタイプ開発の中で、同技術は建築分野向け研究成果PR用のプロトタイプシステムで有効に活用された。

空間共有アプリケーション構築用ライブラリの研究では、空間共有ライブラリの整備が遅れているが、遠隔授業プロトタイプの中では有効に機能した。平成19年度は、ライブラリとしての整備とドキュメント開発に注力する予定である。

知的創造プロセスの支援のためのデータベース開発ではKJ法アプリケーションを軸に開発を進めてきた。平成19年度も開発を継続するが、そのままアプリケーションとして事業化できるのか、その中の要素技術を事業化するのか、あるいは、オープンソースとして活用するかの判断を行う必要がある。

アプリケーション開発については、3種類のプロトタイプ開発と実証試験を実施し、それぞれ成果を得た。このうち、遠隔授業はタイル型ディスプレイでの実証試験として継続する。本社・支店間接続試験は、回線の状況もあり有効な試験が見込めないので中止。建築分野の開発は、事業化のためにも継続を提案する予定である。新規のプロトタイプ開発としては、事業目標として大きなウエイトを持つ通称E黒板のプロトタイプ開発と実証試験を進める。その他、医療分野など、新しい事業機会の調査は継続的に実施する。

また、各サブテーマの統合試験として建築分野向けにIPTシステムの活用模擬試験を実施する予定である。

5 参考資料・参考文献

5-1 研究発表・講演等一覧

通し番号	発表方法	発表雑誌名、講演会名、学会名等	発表者	発表タイトル	発表予定月日
18-001	論文誌	日本バーチャルリアリティ学会論文誌、Vol. 11, No. 3, pp. 403-410	小木哲朗、内野孝哉	動的負荷分散レンダリングを用いたCAVEシステム	2006. 9
18-002	査読付き国際会議	IEEE 20th International Conference on Advanced Information Networking and Applications (AINA 2006), Vol.2, pp. 647-651	Yasuo Ebara, Tetsuya Nabuchi, Naohisa Sakamoto, and Koji Koyamada	study on Eye-to-Eye Contact by Multi-Viewpoint Videos Merging System for Tele-immersive Environment	2006. 4. 18
18-003	査読付き国際会議	the 6th IASTED International Conference on VISUALIZATION, IMAGING, AND IMAGE PROCESSING (VIIP2006) pp. 453-458	Yukio Yasuhara, Naohisa Sakamoto, Yasuo Ebara, Hiroshi Katao, Koji Koyamada	Support System for Estimation of Earthquake Fault Plane within IPT	2006. 8. 30
18-004	査読付き国際会議	2006 ASIAGRAPH, pp. 67-72, Shanghai, China	小木哲朗、酒井満隆	Communication in the Networked Immersive Environment	2006. 6. 23
18-005	外部機関誌論文	東京大学大学院工学研究科 学位論文	宮地英生	問題解決環境としての可視化システムの開発	2006. 8
18-006	口頭発表	ヒューマンインタフェース学会研究会研究報告集 Vol.8 No.2, pp. 61-64	久木元 伸如, 江原康生, ジェイソンリー, 小山田 耕二	タイルドディスプレイを用いた遠隔コラボレーションに関する検討	2006. 6. 7
18-007	口頭発表	ヒューマンインタフェース学会研究会研究報告集, Vol. 8 No. 2	宮地英生、大吉芳隆、酒井満隆、小木哲朗	OpenGL Fusion によるビデオアバタと物体合成の性能評価	2006. 6. 7
18-008	口頭発表	計算工学会講演会論文集、Vol. 11	宮地英生、谷前太基、松尾武洋、大吉芳隆	OpenGL 合成技術によるVR空間でのアプリケーション統合	2006. 6
18-009	口頭発表	WCCM(Los Angels)	Shigeo Kawata, Koichi Shimizu, Hirokazu Nishiura, Hideo Miyahi, Sugiura Hideaki, Yuichi Saitoh	An Intelligent Distributed Problem Solving Environment (PSE) for Scientific Computing	2006. 7. 20
18-010	口頭発表	日本バーチャルリアリティ学会 第2回テレマージョン研究会	宮地英生	Latest Report about CnC Project - Now, Open CABIN Library available !?	2006. 10. 10
18-011	口頭発表	可視化情報シンポジウム、Vol. 26, Suupl. No1., pp. 239-240, 2006	宮地英生、谷前太基、松尾武洋、大吉芳隆	OpenGL Fusion によるCGと工学の統合	2006. 7. 25

18-012	口頭発表	可視化情報全国講演会、 vol.26, Suppl No.:2, pp.67-68	宮地英生, 小木哲 朗, 小山田耕二, 江原康生, 久木元 伸如, 廣瀬通孝	テレ・イマージブ・ カンファレンスのた めの基盤ソフトウェ アの開発	2006.9.27
18-013	口頭発表	日本バーチャルリアリ ティ学会第11回大会論文 集、vol.11, pp.460-463,	宮地英生, 大吉芳 隆, 立山義祐, 小 木哲郎, 江原康 生, 久木元伸如, 小山田耕二, 廣瀬 通孝	OpenGL フェュージョン によるリッチな IPT 空間の構築	2006.9.9
18-014	口頭発表	日本バーチャルリアリ ティ学会 第11回大会, pp.452-455,	江原康生, 久木元 伸如, リージェイ ソン, 小山田耕二	タイルドディスプレ イ環境における高解 像度映像による遠隔 コミュニケーション 実験	2006.9.9
18-015	口頭発表	日本バーチャルリアリ ティ学会 第11回大会, pp.456-459	安原幸生, 坂本尚 久, ノナカジョル ジ, 江原康生, 小 山田耕二	シーングラフと可視 化パイプラインを同 一グラフ上で編集で きる可視化システム	2006.9.9
18-016	口頭発表	日本バーチャルリアリ ティ学会 第2回テレイマ ージョン技術研究会	久木元伸如	タイルドディスプレ イによる大画面提示 装置を用いた知的作 業支援	2006.9.25
18-017	口頭発表	宇宙地球系情報科学研 究会(京都大学)	宮地英生	再コンパイル・リン クなしに3次元情報 を重ねるOpenGLフェ ュージョン	2006.11.29
18-018	口頭発表	第16回設計工学・システ ム部門講演会講演論文 集、pp.343-344	小木哲朗, 左近浩 章	テレイマージョン技 術を用いた遠隔授業 システムの構築	2006.11.17
18-019	口頭発表	日本バーチャルリアリ ティ学会 第11回大会論文 集、pp.404-407	小木哲朗, 村瀬香 織	大画面ハーフミラー を用いた没入型拡張 現実環境の構築	2006.9.9
18-020	口頭発表	日本バーチャルリアリ ティ学会 第11回大会論 文集、pp.408-409	立山義祐, 谷川 智洋, 広田光一, 廣瀬通孝	普及版 CABIN ライブ ラリの概要	2006.9.9
18-021	口頭発表	日本VR学会テレイマ ージョン技術研究会第一回研 究会	立山義祐	広く使われることを 念頭に置いた CABIN ライブラリのオープ ンソース化	2006.5.12
18-022	口頭発表	日本バーチャルリアリ ティ学会 第3回テレイマ ージョン技術研究会	宮地英生	CnC プロジェクト最 新レポート ~平成 18年度総集編~	2007.1.26
18-023	その他 資料	産業用バーチャルリアリ ティ展、東京有明：国際 展示場	(株)フィアラックス	FusionVR を出展。(株) クレッセントブース	2006.6.21-23
18-024	その他 資料	SIGGRAPH 展示会、米国ボ ストン	(株)フィアラック ス、(株)ケイ・ジー・ ティー	FusionVR を出展。(株) フィアラックス、(株) ケイ・ジー・ティー 合同ブース。	2006.8.1-4

18-025	その他資料	ANSYS カンファレンス、東京ホテルメトロポリタン(池袋)	(株)ケイ・ジー・ティー	FusionVR 出展。(株)ケイ・ジー・ティーブース。	2006. 11. 15-16
18-026	その他資料	CollabTech 2006、University of Tsukuba	小木哲朗、林正紘、村瀬香織	デモ展示: Immersive Projection Environments and Tele-immersion Applications	2006. 7. 13
18-027	口頭発表	日本バーチャルリアリティ学会 第1回テレ-immersion 研究会	内野孝哉	動的負荷分散レンダリングを用いた CAVE システム	2006. 5. 12
18-028	口頭発表	ヒューマンインタフェース学会 研究会 研究報告集, Vol. 8 No. 3, pp65-68	酒井満隆、小木哲朗、立山義祐、江原康生、宮地英生	ビデオアバタを用いた多地点間 CAVE コミュニケーション	2006. 6. 7
18-029	口頭発表	日本バーチャルリアリティ学会 第2回テレ-immersion 研究会	酒井満隆	JGNII ネットワークを用いたビデオアバタコミュニケーション	2006. 10. 10
18-030	査読付き国際会議	ACM Symposium Virtual Reality Software and Technology (VRST2006), pp. 189-192	Tetsuro Ogi, Takaya Uchino	Dynamic Load-Balanced Rendering for a CAVE System	2006. 11. 2
18-031	口頭発表	2006 年度メディア情報処理専修コース ビジュアルライゼーション (京都大学)	立山義祐	普及版 CABIN ライブラリの概要	2006. 9. 29

別紙1：SC06出張報告

ハイスピードコンピュータの国際的な祭典 SC06 がフロリダ州 Tampa (タンパ・コンベンション・センター) で 11 月 13-18 日に開催された。登録入場者数は約 7100 名、出展社数 274 で、前年のシアトル (約 1 万名) を下回ったものの盛大に実施された。

1. 最高速マシン

年 2 回発表される TOP500 リストは、リンパックのベンチマークを使って世界の最高速を競うものである。この秋のリストで、第一位は DOE/NNSA/LLNL で、IBM BlueGene/L の 131,072 並列で達成した 280TFLOPS (主記憶 32.768Tbyte) であった。

ベスト 10 を国別で見ると米国 7、スペイン、フランス、日本 (東工大の TSUBAME 9 位) で、ベンダー別に見ると IBM5, Cray, Dell, SGI, BullSA, NEC/Sun である。

かつて栄光を誇っていた地球シミュレータは現在 14 位になっている。

2. アワード

SC では、いくつかの表彰が行われる。

(1) ベストペーパー

“Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters”, K.J. Bowers et al. D.E. Shaw Research.

(2) ゴードンベル賞

“Large-Scale Electronic Structure Calculations of High-Z Metals on the BlueGene/L Platform”, team of Francois Gygi, Univ. of California, ピーク性能を出した論文を書いた論文であった。他、特別賞として、やはり、BlueGene/L を使った量子計算と、日本からは MD-GRAP3 を使った計算をエントリーした理研が選ばれた。

(3) アナリティック・チャレンジ

“テラスケールシミュレーションと可視化を統合した遠隔操作 “カリフォルニア大学など。

(4) ストレージチャレンジ (大規模システム編)

“Gfarm ファイルシステムを使った粒子物理のための高性能データ解析” 筑波大学。

(5) ストレージチャレンジ (小規模システム編)

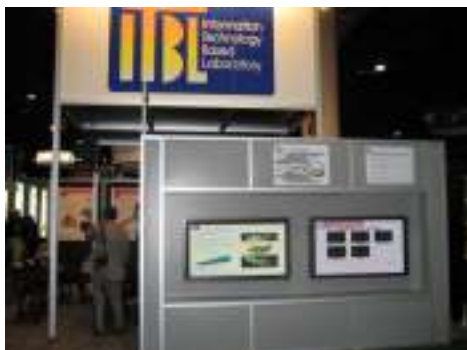
“大規模計算化学アプリケーションのための高速 InfiniBand ディスクアレイへの並列アクセスを使った Trading Memory for Disk” Ames 研究所。

(6) バンド幅チャレンジ

“SECTOR を使った Sloan デジタルスカイ・サーベイデータの転送 “イリノイ大学。8 ギガビット/秒を記録し、ピーク性能は 9.18GBps であった。

3. グリッド

グリッド基盤は米国の TeraGRID、欧州の EGEE(Enabling Grids for E-sciencE) が世界的に運用されているグリッドシステムとして参加。日本では ITBL と NAREGI の参加があった。NAREGI は欧米との相互協力も行っているが日本での運用実績が少なく限定されたグリッド利用に留まっている。日本ではグリッド基盤の開発が続いているが、欧米では研究の中心は既に運用に進んでいる。



I T B Lのブース。防災研のコンテンツがメイン。



バークレイのブース。
“Next Generation Grid Tool”が看板に。

E V Lら。TeraGRID&マルチ表示。



E-Science のブース。
ジョブの運用状況をモニタリング中。

teraGrid: 102Tflops, 15 PetaByte の規模を持つ世界最大のグリッドシステム。シカゴ大学で GIG (Grid Infrastructure Group)がコーディネートしている。米国の主要大学、国立研究所がリソースを提供している。

<http://www.teragrid.org/>

EGEE:32 カ国が参加。2006 年 4 月 1 日からヨーロッパコミッションのファンドで 2 年間の第 2 ステージ運用が始まっている。24 時間稼働。

<http://www.eu-egee.org/>

4. 大画面表示

プロジェクトは少なくなり、フラットモニターによるマルチディスプレイが主流になっている。

アップル社。マッキントッシュプロ上で Ensign が動作している。



タンガステン・テクノロジー社の可視化サーバ。
Chromium をビジネス化している会社。

OAK Ridge は大型ディスプレイを縦に配置して 5 面のマルチスクリーンに。





TeraGRID ブース。



大阪大学 (SAGE を使っている)

5. 大規模可視化

マンチェスター大学では並列AVS (PST) のデモを医療データで実施。

Mercury 社は Cell をグラフィックスに利用 (右下)。Amira インタフェースもあるが 2D のみで 3次元インタフェースの予定は無いとのこと。



SUNブースでは 197 億△の大規模可視化事例が東工大のTSUBAMEとともに紹介された (下)。



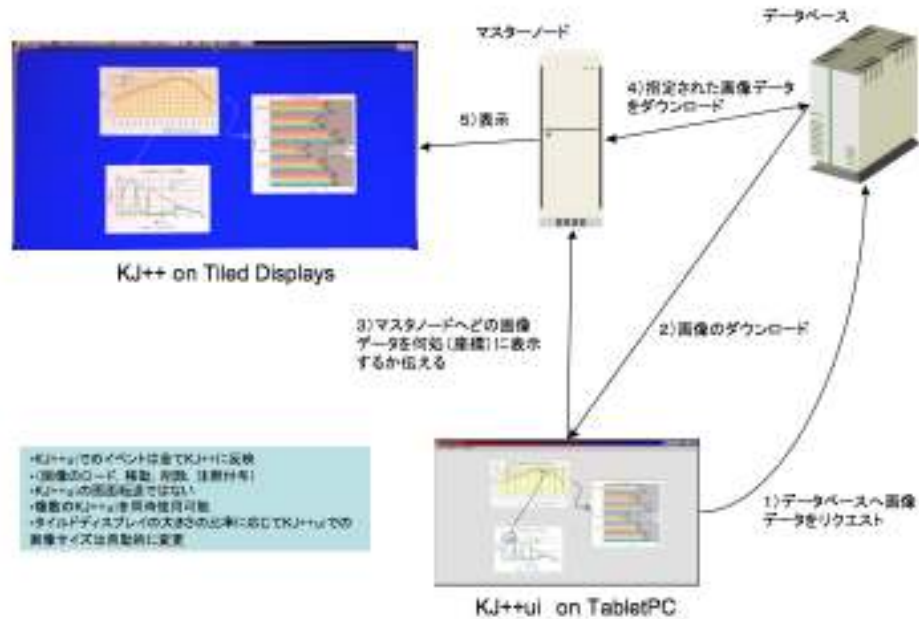
6. バーチャルリアリティ

ボストン大学は偏光立体視を (左下)。他、ITBLでも立体視映像を上映していたが、昨年よりは立体視は減っていた。右下は、両手にサイコロを持ち、それをカメラ撮影で方向を検知しながら、医療データの任意断面を取り出すVRインタフェースを操作している様子。斜めのディスプレイの向こう側に手を入れると、画面に手が登場する。



別紙2：KJ法システム構成とクイックスタートガイド

- システム構成



- REQUIREMENT

KJ++
Linux
Xdmx
PostgreSQL
QUANTA
Apatch
KJ++ui
Windows TabletPC Editioin

- Quick Start

各マシンは multiuser mode に移行

```
%telinit 3
```

タイル型ディスプレイの各ノードでXサーバ起動

```
% xinit
```

各 display ノードのターミナルで xhost を 設定

```
% xhost +
```

master node で dmx を 起動

```
% startTiledmx_8
```

5 master node でウインドを起動

```
% mwm &
```

6. タイル型ディスプレイ上で右クリックで新規ターミナルを開く.

7. KJ++をインストールしたディレクトリに移動しKJ++を起動

```
%kjpg
```

- データベース

データベースはPostgreSQL を用いるので

<http://www.postgresql.org/>

からダウンロードし、インストールする

PostgreSQL データベースのインストールでは、新しく作った postgres ユーザで行う。インストールのパスは/usr/local/pgsql/になる、データベースのデータが./data フォルダに置いてある、設定は./data フォルダの下の pg_hba.conf ファイルによるデータベースのアクセスセキュリティの設定を行う。

すべてのマシンからアクセスできるようになるのは

```
Host all all 0.0.0.0 0.0.0.0 trust
```

を挿入する。

下記のようなデータ構造になっている。

テーブル Vring はデータ用

```
create table vring (  
    no serial primary key,          -- データ番号  
    cdate date,                    -- 作成日  
    fname text,                    -- ファイル名  
    fsize integer,                 -- ファイルサイズ  
    ftype text,                    -- 画像 MIME type  
    description text,             -- 説明  
    by text,                       -- 作成者  
    file oid,                      -- ファイルデータ (バイナリ)  
    title text                     -- コメント  
);
```

テーブル parameter はパラメータ用

```
create table parameter (  
    parameter date,               -- パラメータ  
    floatvalue float8,           -- 値  
    id text                       -- データ番号に関連する  
);
```

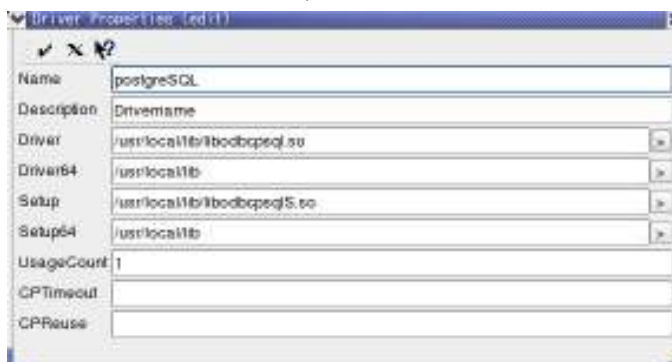
```
grant all on vring to nobody;
```

```
grant all on vrreply to nobody;
```

●ODBC

ODBCConfig コマンドで ODBC 管理する画面が出てくる。

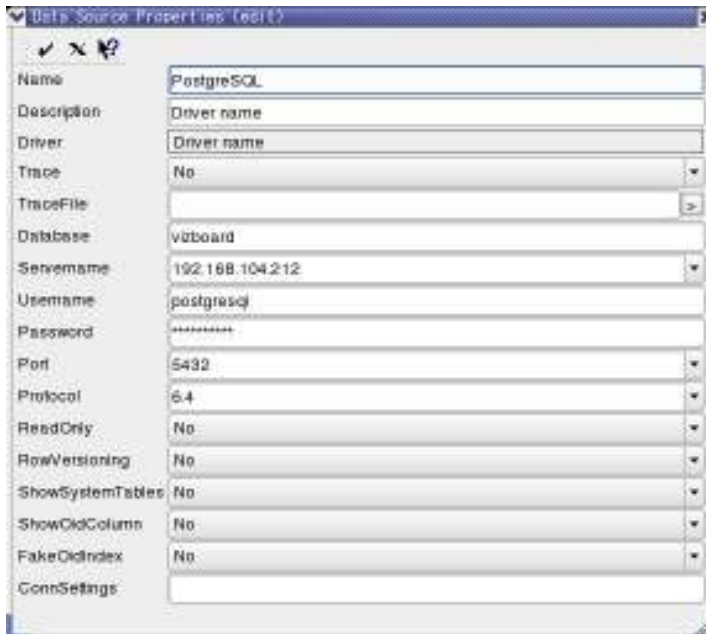
先に ODBC のドライバを Add する。



Name のところが任意でよい。

Driver と Setup のところに Postgresql をインストールパスを正確に入力する。

UserDSN を設定する



Driver : DriverProperty の名前と同じ名前で設定する

Database : Postgresql のデータベース名を入力する

Servername : データベースサーバの IP アドレスを入力する

Username : データベースのオーナーユーザの名前を入力する. KJ++では

補足

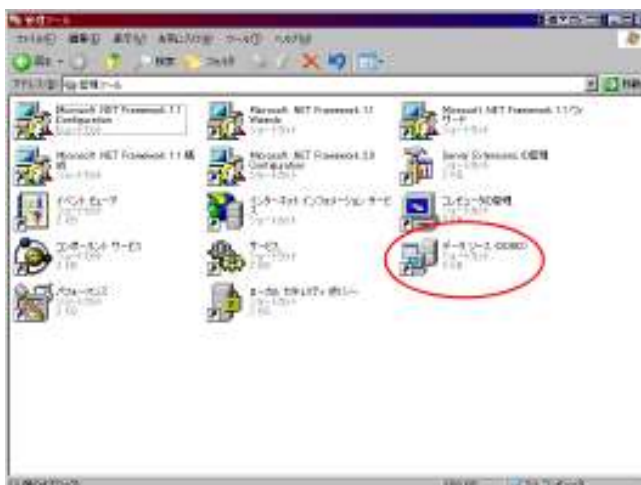
Linux 側のアクセス関数は

```
SQLConnect( Hdbc , (SQLCHAR*) dns , SQL_NTS< (SQLCHAR*)uid, SQL_NTS, (SQLCHAR*)pid, SQL_NTS);
```

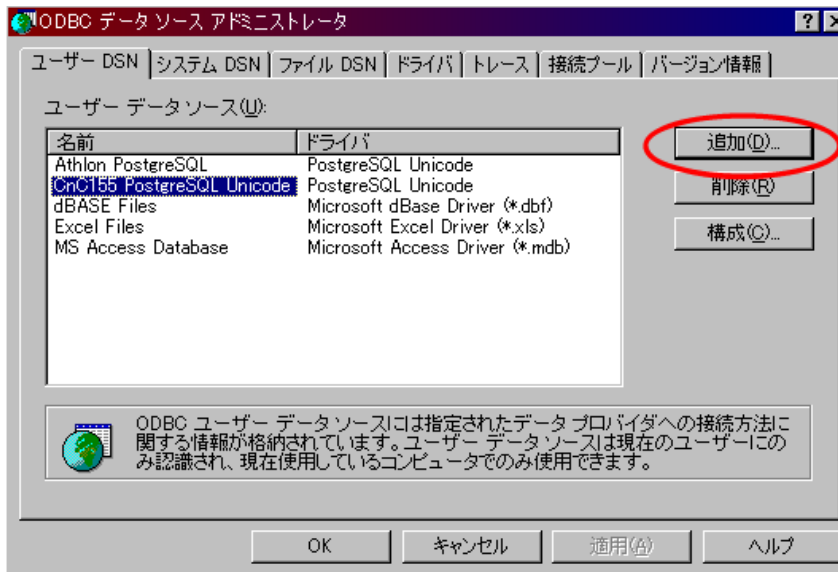
この中の dns は UserDNS による設定した名前のことになる

• Windows 側の設定

スタート→設定→管理ツールからデータソース (ODBC) を選択



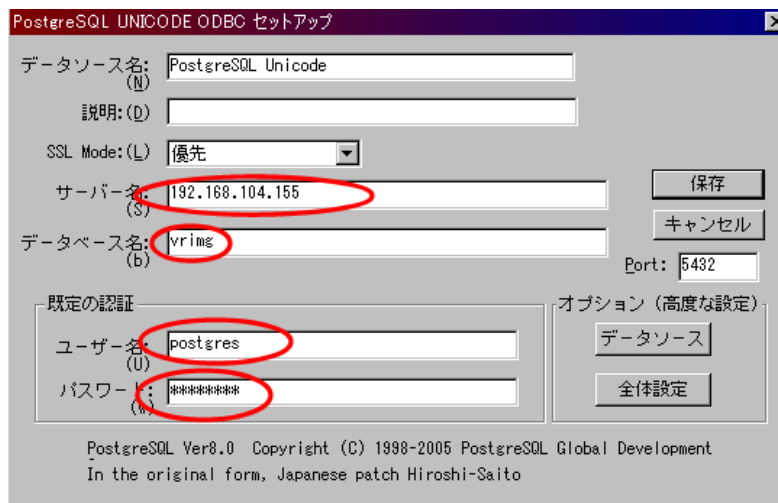
「追加ボタン」として新しいドライバを登録する



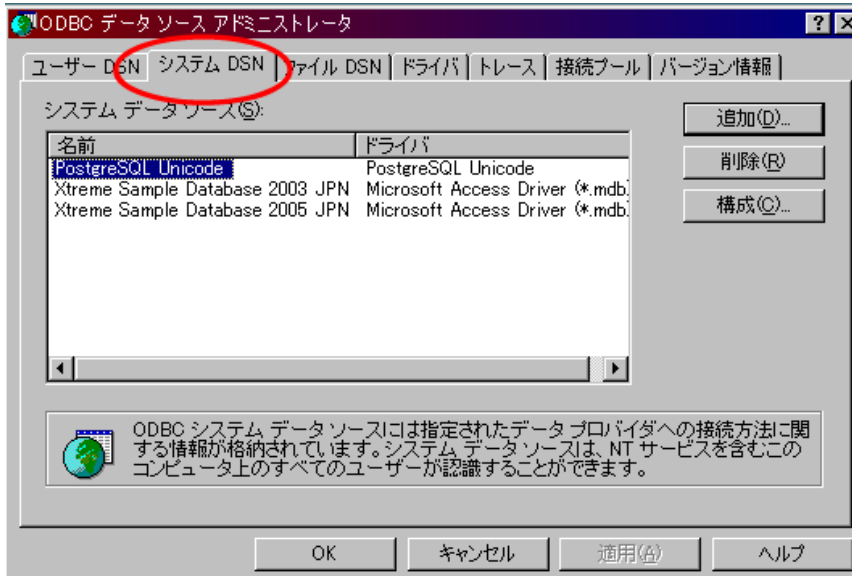
「PostgreSQL Unicode」を選択する。
無い場合には <http://pgfoundry.org/projects/psqlodbc/> を参照。



PostgreSQL UNICODE ODBC のセットアップを行う。
サーバー名 : データベースが稼働しているマシン名もしくは IP アドレスを指定
データベース名 : アクセスするデータベースを指定
既定の認証 : データベースにアクセスするユーザ名をパスワードを指定する



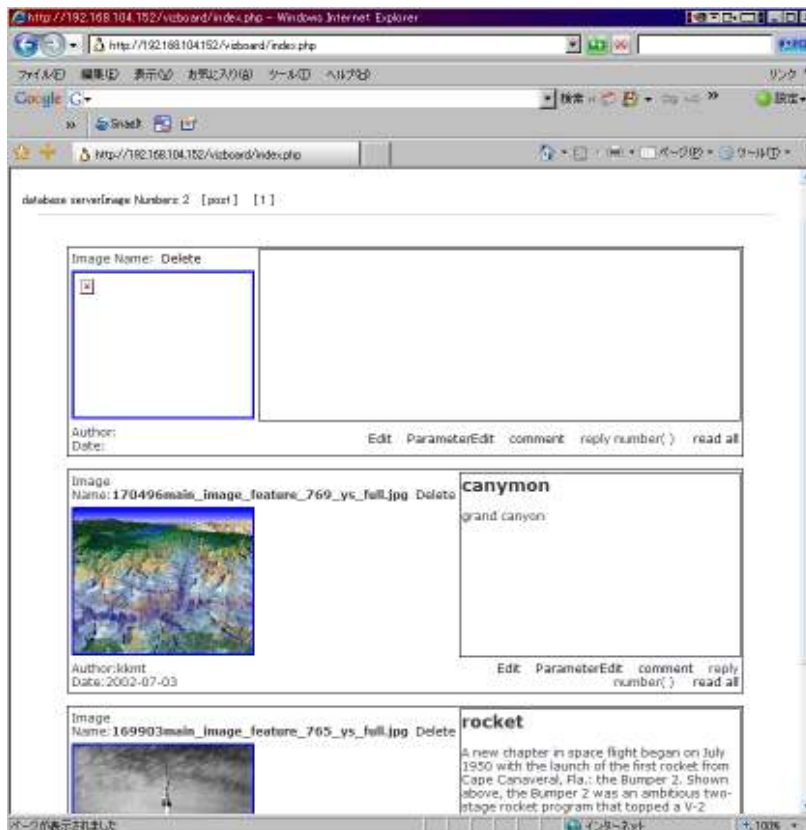
システム DNS タブを選択肢データソースを選択する。



- データベースへの登録

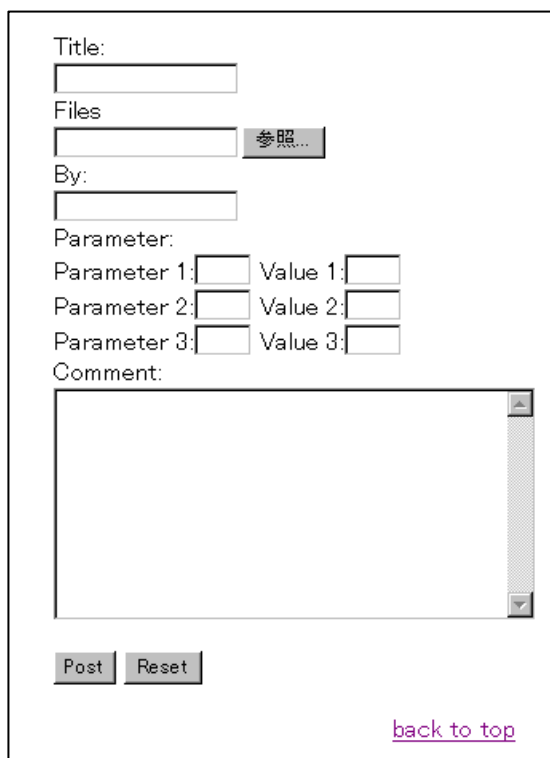
データベースへのデータの登録・削除等はブラウザを用いて行う。

各画像にはコメントを書き込む事が出来る。このコメントは KJ++ui で参照するときデータを読みやすく説明するとき役立つ。パラメータは検索時に用いる



画像の登録

[post]をクリックし、登録画面を呼び出す。



The screenshot shows a web form for image registration. It includes the following elements:

- Title:** A single-line text input field.
- Files:** A file selection input field with a "参照..." (Browse...) button.
- By:** A single-line text input field.
- Parameter:** A label for the parameter section.
- Parameter 1:** A text input field followed by "Value 1:" and another text input field.
- Parameter 2:** A text input field followed by "Value 2:" and another text input field.
- Parameter 3:** A text input field followed by "Value 3:" and another text input field.
- Comment:** A large multi-line text area with a vertical scrollbar on the right.
- Buttons:** "Post" and "Reset" buttons at the bottom left.
- Link:** A "back to top" link at the bottom right.

title 画像のタイトル

File 登録する画像を指定

By 登録者の名前

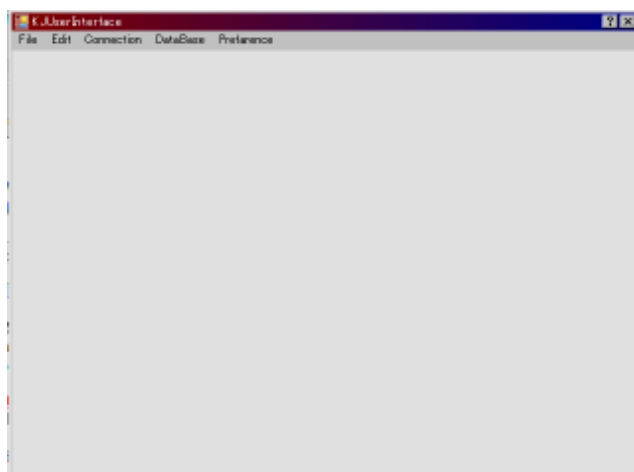
Parameter 検索時に用いるパラメータ. Parameter と Value を組みにして登録する

Value そのパラメータの値. 例えば. 「温度」はパラメータで, 35℃は Value となる

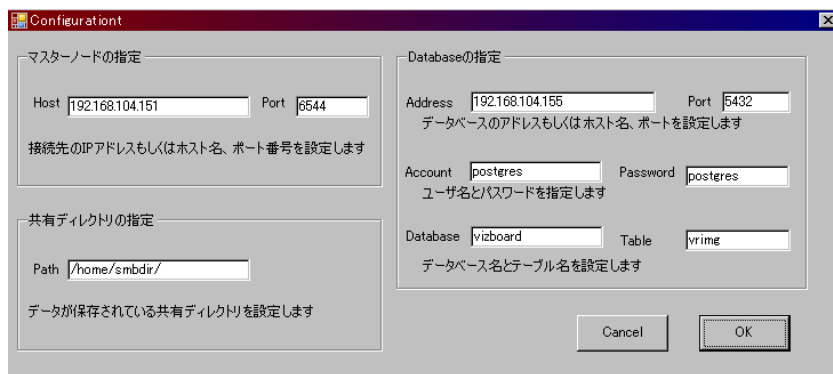
Comment 画像に対するコメント

- KJ++ui 操作方法

KJ++ui を起動する. 同じディレクトリのファイルの中身は変更してはいけない.



Preference メニューで接続先等を指定する.



マスターノードの指定 KJ++のマスターノードの IP とポート番号を指定する
共有ディレクトリの指定 KJ++が参照する samba のディレクトリを指定する.
注意!) KJ++ui では無く, マスターノードで動く KJ++が参照するディレクトリを指定する

Database の指定

Address データベースが稼働しているマシンの IP アドレス

Port データベースのポート番号.

Account データベースにアクセスするアカウント名

Password データベースにアクセスするアカウント名に対するパスワード

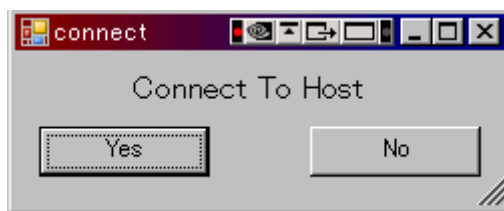
Database 参照するデータベース名

Table 参照するテーブル

補足)

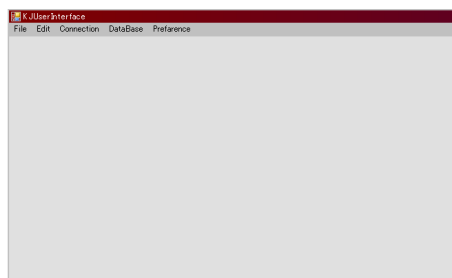
KJ++ui は内部で ODBC によるデータベースのアクセス (OS で設定) と Npgsql による 2 つのアクセス方法を有している. ODBC によるアクセスでは BLO (Binary large Object) の参照を WindowsSQL サーバでしか出来ない. そこで, Npgsql を用いた

マスターノードで KJ++ を実行し, KJ++ui から KJ++へ接続する
KJ++ui の Connect メニューをクリックし, 接続ダイアログを開く

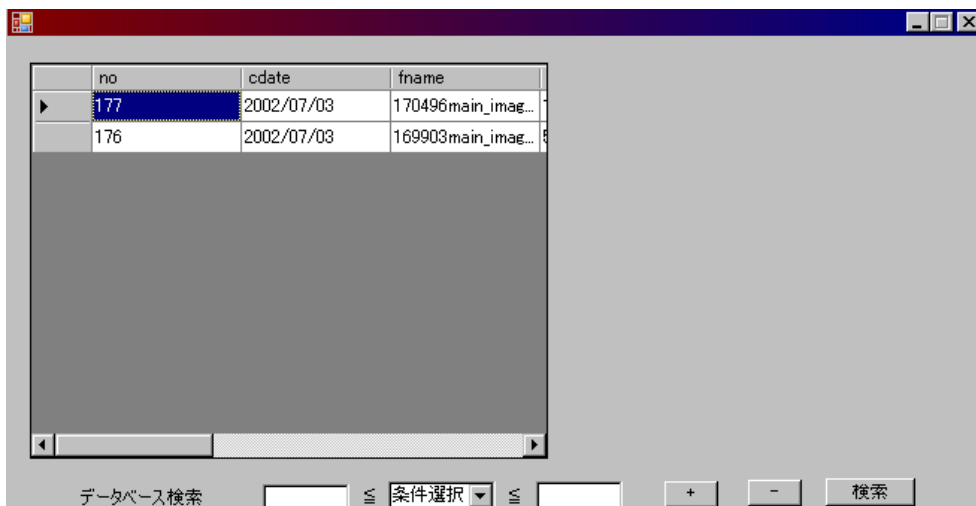


Yes ボタンをおして KJ++へ接続する.

接続が確立されれば, KJ++ui のウインドウの大きさが, KJ++ui を実行しているマシンのディスプレイの大きさの中で KJ++と同じ縦横比になるように modify される.

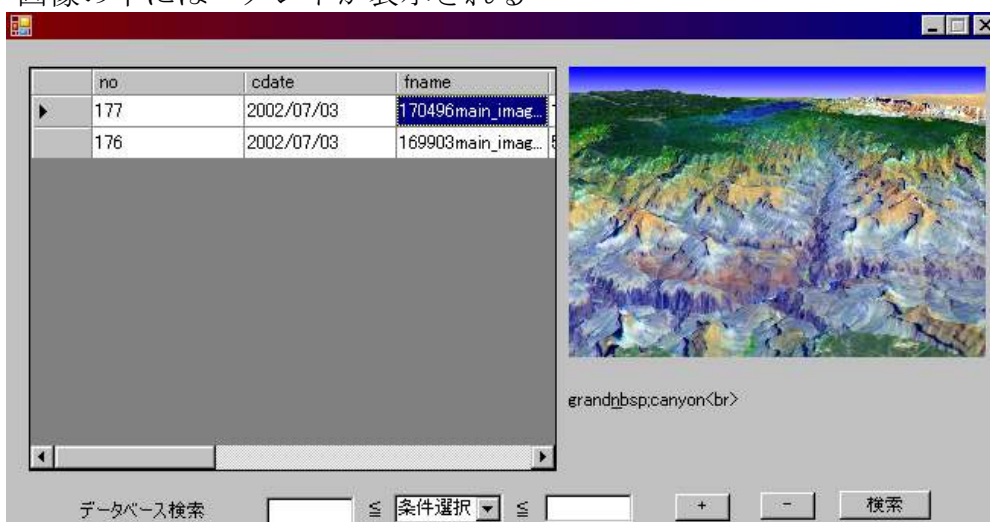


Database メニューをクリックし，データベース一覧を表示する



条件付き検索を行うときは，条件検索から任意の Parameter を選択し，上限下限の Value を与えて検索する．検索条件はブラウザで登録した Parameter が読み込まれる．

ファイル名のクリックによって一覧の右側に画像が表示される（上手く行かない事多々あり）．画像の下にはコメントが表示される



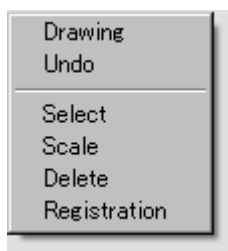
右側に表示された画像をタイル型ディスプレイに表示する場合には，画像を KJ++ui のメインウィンドウにドラッグ&ドロップすることでタイル型ディスプレイに表示される．タイル型ディスプレイに表示される大きさは実サイズで表示されるが，KJ++ui への表示は KJ++ui と KJ++のウィンドウに比率に合わせて縮小される．

データベース一覧のウィンドウを消すときは CloseButton(右上の X ボタン)をクリックする．

Samba に登録されている画像を読み込むときは samba で共有されているフォルダを開いて画像を KJ++のウィンドウへドラッグ&ドロップしてタイル型ディスプレイに表示する．

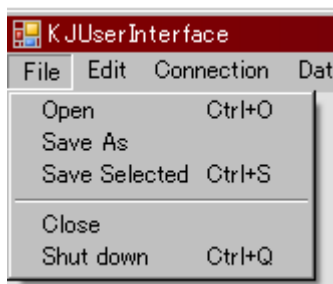
画像の位置を変更するときは KJ++ui の画像をカーソルで動かせばタイル型ディスプレイでの表示位置も変わる．

KJ++ui 上で右クリック (tabletPC の場合は長押し) でメニューが表示される. このメニューはプルダウンメニューの Edit と同じである.



- Drawing 注釈描画モードへ切り替える. このとき画像の位置変更は出来ない. 画像の位置を変更する場合は再度 Drawing を選択 (チェックを外す) する.
- Undo 注釈の undo を行うつもりだけど未実装
- Select 指定した画像の削除や登録を行うときに用いる. Select を選択し, 画像をクリックした場合, 画像に枠が表示される. 枠の着いた画像のみが後述の機能の対象となる.
- Scale 画像の拡大・縮小をやりたかったけど, 未実装
- Delete select で選択された画像を消す. Database の中の画像は消えない.
- Registration Samba で参照した画像をデータベースに登録する場合に用いる.

現在の作業状況の保存や終了は File メニューから選択する



- Open 画像の位置と座標, 注釈を記録された set ファイルを読み込む
- Save As 現在の作業状況を保存する. 作業があるのは. set 拡張子で保存される. 保存ダイアログで拡張子はつけなくて良い.
- Close KJ++ui のみを終了する
- Shut down KJ++ui, KJ++共に終了する.