

仮想空間における複数の立体映像を用いた TV 会議システムの開発

折田 真一 (310201110)

名古屋大学大学院 人間情報学研究科 物質・生命情報学専攻

1 はじめに

近年、マルチメディアに関するハードウェア及びソフトウェアの性能向上に伴い、VR (Virtual Reality) が比較的身近なものになってきた。コンピュータの低価格化などにより、従来では困難であったリアルタイム 3D レンダリング処理が市販されているパソコンで十分まかなえるようになってきたからである。そのような流れを受け、2003 年 3 月に "3D コンソーシアム"[1] が設立され、3D を扱った多くのアプリケーションが登場した。NTT-X 社の "3DFieldExplorer"[2]、3DNA 社の "3DNA デスクトップ"[3] などが記憶に新しい。さらに、裸眼立体視液晶ディスプレイを搭載したノート PC [4, 5] も一般に市販され、一般人に対する VR への敷居は徐々に低くなってきている。

このように、今後、マルチメディアインタフェースが 3D へ移行するに伴い、従来とは異なった三次元画像を活用したコミュニケーションツールの必要性が高まることが予想される。なぜなら、3D インタフェースは 2D インタフェースに比べ、臨場感、情報伝達能力、そして表現力において優れており、またエンターテインメント、医療、そして教育といった多方面での応用が考えられるからである。

そのような VR を利用したコミュニケーションツールのひとつとして、仮想空間における複数の立体映像を用いた TV 会議システムが考えられる。これは、仮想空間における複数の仮想立体 TV を通じて、遠隔地にいる相手とコミュニケーションを行なうシステムであり、ユーザーは仮想空間内で仮想立体 TV を覗くことで、まるでその場にいるかのように相手と会話することができる。

そこで本研究では、仮想空間における複数の立体映像を用いた TV 会議システムの開発を行なった。また、従来の VR システムは、高価なデバイスやソフトウェアを必要とし、実現性や可搬性が低かったため、比較的汎用性の高いシステムの構築を目指した。

2 仮想空間における立体映像の原理

本システムで使用した仮想空間における立体映像を実現する原理について説明する。

図 1 は、仮想空間における立体映像の原理を示した図である。まず、PC 上において仮想空間を構築し、視差を持った視点及び立体映像を表示させるオブジェクト (プリ

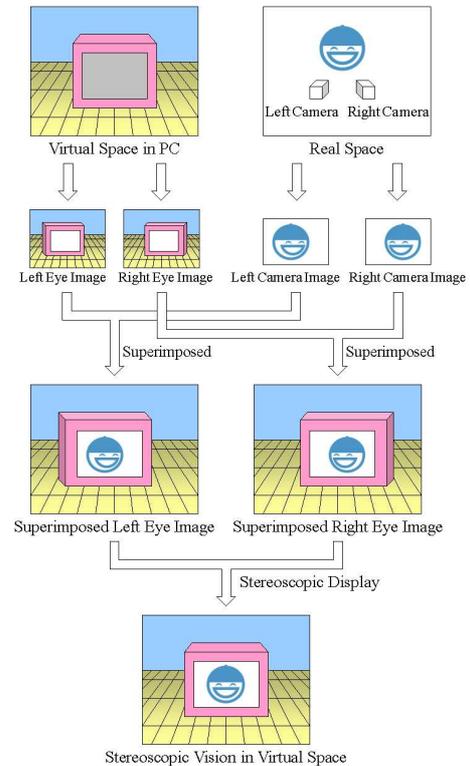


図 1 仮想空間における立体映像の原理

ミティブ)を作成する。それとは別に、実空間の対象物を 2 台のカメラで撮影し、各カメラから画像を取り込む。そして、仮想空間内から切り出された左眼用と右眼用の画像に描かれたオブジェクト上に、カメラから取り込んだ画像をそれぞれ貼り付け、この視差画像を立体視する。すると、観察者は仮想空間に対して立体感を知覚でき、さらにオブジェクトに投影された実空間の映像に対しても立体感を知覚できる。

しかし、これを実現するには、3D レンダリング方法に大きく依存する。最も重要な要素は、仮想空間内の視点に対して描画するオブジェクトを指定できるかどうかである。なぜなら、仮想空間内で立体映像を見るには、左眼画像、右眼画像のそれぞれにしか描画されないオブジェクトが必要だからである。

Java3D では、オブジェクト (プリミティブ) に対して 2 つの属性を指定でき、各視点に対してどちらの属性のオブジェクトを可視化するかを決定できる。この機能を利用することで、仮想空間における立体映像を実現できる。

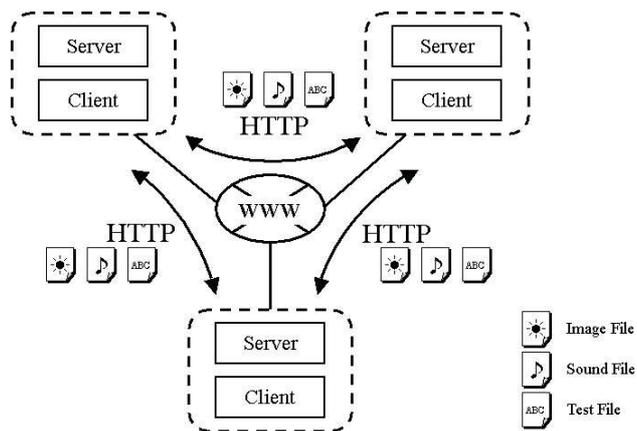


図 2 システム構成

3 開発したシステム

3.1 システム構成

本研究で開発した仮想空間における複数の立体映像を用いた TV 会議システムの構成を図 2 に示す。本システムは、画像・音声サーバーアプリケーション、TV 会議クライアントアプリケーションから構成されている。なお、入力インターフェイスにはマウス、マイク、キーボード、そして USB カメラといった標準的なデバイスを、出力デバイスには立体視ディスプレイ（又はその代替となるデバイス）、スピーカー（ヘッドフォンなど）を使用する。

3.2 Java による USB カメラ画像の獲得

本研究では、汎用性の高いシステムを構築するにあたり、Java による USB カメラの画像の獲得を試みた。本システムでは、javax.usb [6] 及び JMF [7] を用いて USB カメラの画像を獲得している。これらについて、以下で説明する。

javax.usb を利用したドライバの開発 本研究では、javax.usb を用いて USB カメラドライバ “usbcamera パッケージ” を作成した。usbcamera パッケージは、Pure Java (Java のみで記述されている) の USB カメラドライバであるため、ドライバレベルからアプリケーションレベルまで、完全に Java ベースで記述できるようになり、非常に汎用的なシステムを構築することができる。

JMF の利用 JMF はメディアコンテンツの取り込み、再生、ストリーミング、そしてコード変換などを取り扱うための API であり、J2SE のオプションとして提供されている。本システムでは、JMF によって USB カメラの画像を獲得することも可能である。

3.3 画像・音声サーバーアプリケーション

本研究で開発した 画像・音声サーバーアプリケーションの構成を図 3 を示す。本サーバーアプリケーションは、コンピュータに接続された USB カメラの画像及び接続されたマイクの音声をストリームとして生成し、HTTP プロト

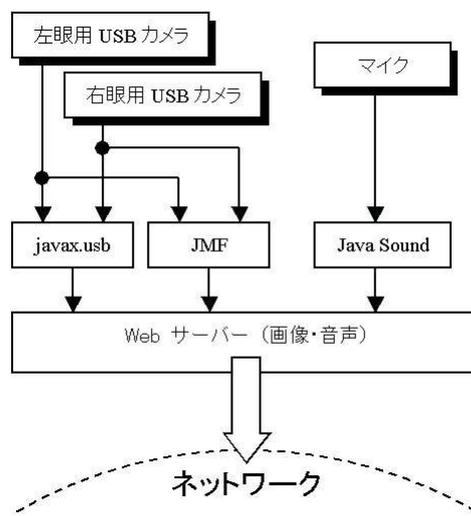


図 3 画像・音声サーバーアプリケーションの構成

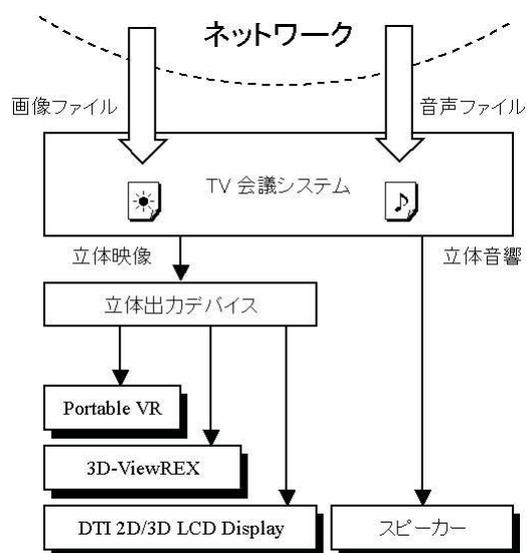


図 4 TV 会議クライアントアプリケーションの構成

コルにより送信するサーバーアプリケーションである。

画像生成 サーバーに接続された USB カメラからの画像情報は javax.usb、もしくは JMF によって構成された画像生成モジュールに送られる。そこで、JPEG ストリームが生成され、サーバーへ渡される。この時点から、USB カメラの画像は HTTP プロトコルによりダウンロード可能となり、次の画像が生成されるまで公開を続ける。

音声生成 サーバーに接続されたマイクからの音声情報は、Java Sound API によって構成された音声解析モジュールに常に送られる。このモジュールはマイクから入力される音声情報を常に監視し、音声の有無を解析する。音声が発見されると、その音声の開始から終了までの区間を音声ストリームとして生成し、サーバーへ渡す。その後、HTTP プロトコルによりダウンロード可能となり、その音声期間だけ公開を続ける。

3.4 TV 会議クライアントアプリケーション

本研究で開発した TV 会議クライアントアプリケーションの構成を図 4 に示す。

本クライアントアプリケーションは、クライアント PC 上で仮想空間を構築し、サーバーから逐次ダウンロードされた画像及び音声ファイルをもとに、仮想空間において立体映像及び立体音響を提供するアプリケーションである。

なお、本クライアントは立体視可能な映像を提供する機能を持ち、立体映像を体験するには立体視用のディスプレイやメガネを、立体音響を体験するにはステレオスピーカーを必要とする。本クライアントアプリケーションでは、下記の 3 つの立体視用デバイスに対応している。

- 偏光プロジェクタ
- プリズムメガネ
- Parallax Illumination 方式の 3D 液晶ディスプレイ

また、操作パネルを図 5 に示す。操作パネルでは、クライアントアプリケーションの設定を行なう。設定は、User Data (接続するサーバーの情報)、GUI Type (GUI のタイプ)、Screen Type (スクリーンのタイプ)、Separate & Ratio Type (映像とアスペクト比のタイプ)、Physical Parameter (スクリーンとスピーカーの設定等) が行なえる。

3.5 システムの特徴

本システムの特徴を以下に記す。

Pure Java アプリケーション 本システムでは、Java から USB カメラの画像を獲得するために、本研究で作成した USB カメラドライバ "usbcamera パッケージ" 及び JMF を用いた。また、本システムの他のモジュールにおいても完全に Java によって作成されているため、極めて汎用性が高く、また統一的なインターフェースを提供可能である。

容易なシステムの導入 本システムでは、立体視用デバイスを除き、標準的なデバイスを使用する。また、数種類の立体視用デバイスに対応しており、システム導入時にユーザー側において適当な立体視用デバイスを選択することが可能である。さらに、マルチプラットフォームを実現しており、実際に Windows, Linux, Macintosh (クライアントのみ) システムでの動作を確認した。

標準的なプロトコルを使用 本システムでは、サーバ - / クライアント間の通信プロトコルに、HTTP プロトコルを使用している。これは、WWW でのファイル転送に使用されているプロトコルであり、きわめて標準的なプロトコルである。また、画像フォーマット (JPEG) 及び音声フォーマット (AU, WAVE, AIFF) においても、標準的なものを採用しており、すでに利用されている画像、音声サーバーによる代用も可能である。

仮想空間における立体映像及び立体音響 従来の VR システムでは、視覚対象は全て仮想的なオブジェクトばかり

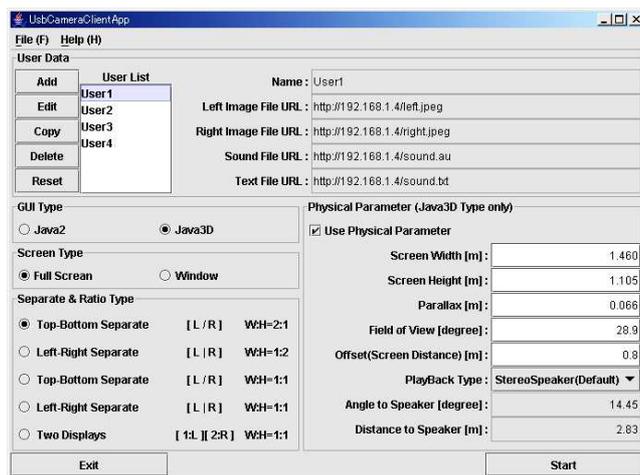


図 5 操作パネルのスナップショット

であった。また、写真を仮想空間内のオブジェクトに貼り付け、リアリティを向上させたり、TV のように演出する試みも数多く存在したが、仮想空間において実空間までも立体的に知覚しようとする試みは、調査した限りでは全くなかった。さらに、VR システムには視覚的に演出するものが多く、聴覚的な演出、つまり立体音響を提供するシステムは少なかった。本研究で作成したシステムは、従来のシステムにはない仮想空間における複数の立体映像及び立体音響を提供する。

4 実験

本研究で開発した、仮想空間における複数の立体映像を用いた TV 会議システムを用いて、実際に TV 会議を行なった。

4.1 実験方法

実験方法及び実験環境を以下に記す。

実験はローカルエリアネットワーク内で行ない、クライアント PC 1 台及びサーバー PC 4 台の計 5 台の PC を使用した。クライアントアプリケーションを実行するには、比較的高い性能が要求されると思われたため、使用する PC の中で最も性能が高い PC を選択した。

4.2 実験結果

本実験において、TV 会議システムにより生成された映像のスナップショットを図 6 に示す。(a) GridLayout は仮想 TV を格子状に配置した時、(b) H-RingLayout は水平リング状に配置した時、(c) V-RingLayout は垂直リング状に配置した時のスナップショットである。各図に表示されている立方体のオブジェクトが立体映像を写し出す仮想 TV であり、それらの上にある立体文字がサーバー名を表している。画面左上には、マウスで選択された TV の情報が空間内に表示され、サーバー名、左眼画像 URL、右眼画像 URL、画像状態、音声状態を確認できる。画面右下には、

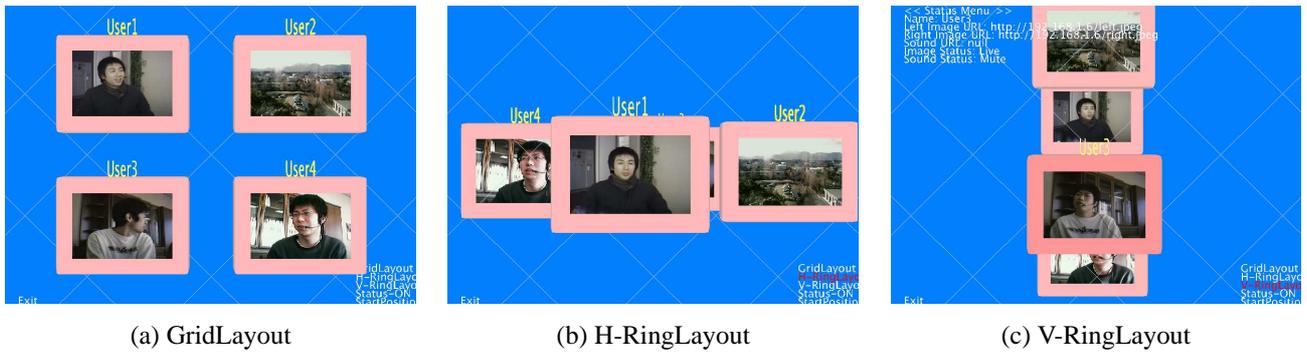


図 6 TV 会議システムによって生成された映像のスナップショット



図 7 立体視用デバイスによる実験の様子

操作コマンドが空間内に表示され、TV の格子状配置、水平リング状配置、垂直リング状配置、TV 情報の表示切替、視点の初期化をマウスによって指定できる。

また、立体視用デバイスによる実験時の様子を図 7 に示す。左から偏光プロジェクタ、プリズムメガネ、Parallax Illumination 方式の 3D 液晶ディスプレイを使用した時の様子を示しており、どのデバイスを使用しても仮想空間において複数の立体映像及び立体音響を確認できた。

5 まとめ

本研究では、仮想空間における複数の立体映像を用いた TV 会議システムの開発を行った。

本システムは、画像・音声サーバーアプリケーション及び TV 会議クライアントアプリケーションから構成される。サーバーアプリケーションは、サーバーに接続された 2 つの USB カメラからの画像及びマイクからの音声をストリームとして生成し、HTTP プロトコルで公開するサーバーである。一方、クライアントアプリケーションは、サーバーからダウンロードされた画像及び音声ファイルをもとに仮想空間において立体映像及び立体音響を提供するシステムである。

なお、システムのコーディングには、高い汎用性を持たせるために Java のみを使用した。また、USB カメラの画像の獲得には、javax.usb を用いて作成したドライバ及び

JMF を用いた。さらに、標準的な入力デバイス、数種類の立体視用デバイスに対応させ、本システムの導入を比較的容易にできるようにした。

最後に、本システムを用いて実際に TV 会議を行なった結果、仮想空間において複数の立体映像及び立体音響を確認することができた。従って、本システムで使用した手法を用いることで、仮想空間において複数の立体映像及び立体音響を実現できることが分かった。

参考文献

- [1] 3D コンソーシアム
<http://www.3dc.gr.jp/>
- [2] 3D ウェブ検索 (NTT-X)
<http://goo.ntt-infolead.net/>
- [3] 3DNA デスクトップ (3DNA)
<http://www.3dna.ne.jp/>
- [4] 3D 液晶搭載ノートパソコン (SHARP)
<http://www.sharp.co.jp/products/pcrd3d/>
- [5] 3D 液晶搭載ノートパソコン (NEC)
<http://121ware.com/lavie/s/>
- [6] javax.usb
<http://sourceforge.net/projects/javax-usb/>
- [7] Java Media Framework API
<http://java.sun.com/products/java-media/jmf/>