

半球型スクリーンを用いた分散レンダリングシステム

鈴木靖子[†] 原田雅之[†] 緒方正人[‡] Jeroen van Baar[§]

[†]三菱電機株式会社 情報技術総合研究所

[‡]三菱プレシジョン株式会社

[§]三菱電機米国研究所

視野全体に広がる没入感のある映像世界を身近に体験できるシステムとして、半球型のドームスクリーンによる映像システムがある。近年、ドーム専用のプロジェクタに代わり、複数の汎用プロジェクタを用いて大型のドームスクリーンに高精細な映像を投影するシステムが提案されているが、コンテンツはドーム専用のものがほとんどである。今回われわれは、OpenGL で記述された平面単一スクリーン用のコンピュータグラフィックスコンテンツを、リアルタイムで変形し分散レンダリングをするシステムを開発した。これにより、既存の CAD などのコンピュータグラフィックスコンテンツをそのままドームスクリーン上に投影することが可能となり、幅広い応用が期待できる。

Distributed Rendering System for Hemisphere Screen

Yasuko Suzuki[†] Masayuki Harada[†] Masato Ogata[‡] Jeroen van Baar[§]

[†]Mitsubishi Electric Corp., Information Technology R&D Center

[‡]Mitsubishi Precision Co., Ltd.

[§]Mitsubishi Electric Research Laboratories, Inc.

Hemispherical dome screen display systems offer immersive visual experiment by covering wider audience's view compared to flat screens. Recently, dome screen display systems with multiple off-the-shelf projectors instead of special projectors for the dome screen are suggested, but contents for them are made for the dome screen only. We have developed a distributed rendering method which projected OpenGL-based CG contents made for flat screens onto the dome screen correctly by distributing and undistorting images in real-time. This method allows us to display various CG contents assets, such as existing CAD system, without modification and therefore apply dome screen display systems to a wide range of fields.

1. 背景と目的

半球型ドームスクリーンは平面スクリーンと比べてより広い視角をカバーでき、映像世界への圧倒的な没入感が得られるため、臨場感や迫力が強く求められるシミュレーションやエンターテインメント分野での大画面映像システムに用いられている。近年、ドームスクリーン専用の高価なプロジェクタを使う方式に対し、汎用プロジェクタを複数組み合わせ高精密映像を実現する方式[1][2]も提案されており、比較的安価なシステムが実用化されている。これらの方式により汎用プロジェクタを用いたドームスクリーンへの重畳投影は可能となったが、投影するコンテンツに関しては、ドームスクリーン専用コンテンツを作成したり、平面用の既存コンテンツに修正を加えてドームスクリーン用に変換したりするが多い。この理由として、ドームスクリーン独自の問題、およびプロジェクタを複数化し分散レンダリングを行う場合に生じる問題がある。

ドームスクリーン独自の問題は、曲面スクリーンに平面コンテンツを投影した場合の歪みである。これは例えば、直線を曲面スクリーンに投影するとスクリーンの曲がりに沿って直線も曲がって投影され、直線として見えなくなってしまう問題である。この問題を解消するためには、投影した状態で直線に見えるよ

うにあらかじめ歪めた曲線を投影する必要がある。また、分散レンダリングを行う場合に生じる問題として、複数のプロジェクタに投影画像を適切に割り当てる手段が必要である。

これらの問題点に対し、あらかじめ歪んだ画像を生成しておいて投影する、ドームスクリーン専用のフォーマットに変更する、コンテンツを分散レンダリング用に複数に分割するなどの手段が用いられているが、いずれの場合にも専用のコンテンツを作成することとなる。

そこでわれわれは、OpenGL^{*}を用いて記述されている平面用の3次元コンピュータグラフィックス(3DCG)コンテンツを、そのままドームスクリーン上に歪みなく表示するシステムを実現する。また、本システムは複数の汎用プロジェクタを用いて半球状の3DCGを高精密に表示できる。これにより、既存の平面用コンテンツ資産をそのまま利用できるとともに、幅広い分野で活用できる。

本稿では半球型ドームスクリーンシステムの概要と、本システムの分散レンダリング技術および性能評価について述べる。

* OpenGLはSilicon Graphics, Inc.の登録商標である。

2. 先行研究

2.1 自動調整

先にも述べたように、複数の汎用プロジェクタ映像の一部を重畳してつなぎ合わせ、ドームスクリーンに1つの映像を投影するためには、投影画像が重なり合う部分を高精度に検出し、投影画像間の位置合わせを行なう必要がある。さらに、平面画像を球面上に投影することによって生じる画像歪量を求め、同時に補正しなければならない(図1)。この対策として、あらかじめスクリーンに対してある位置に固定されたプロジェクタ画像の投影状態をシミュレートし、プロジェクタの位置をその位置になるように調整する手法[2]なども提案されている。一方われわれは、プロジェクタの投影画像を認識して調整する手法を用いる。この手法は、ドーム型スクリーンに投影したキャリブレーション画像を2台のカメラで撮影して、2次曲面に画像を投影した際の射影変換に基づいて相互の調整パラメタを自動的に求める自動調整手法[1]である。以下にこの手法の概要を記述する。

スクリーンが平面の場合には、平面射影変換(Homography)を用いる手法がよく知られている。これは、ある平面上に存在する点を2つの視点から見た場合、2つの視点の射影座標系上に対する点の投影画像座標が、一般に式(1)なる関係があることを利用する[3]。

$$x' = Hx \quad (1)$$

ここで、 x' 、 x はそれぞれの視点における点の射影座標、 H は平面射影変換(Homography)とよばれる座標変換である。この関係式により2つの視点間の関係が求められるため、複数のカメラ画像から同じ画像特徴点を抽出することでカメラ画像のつなぎ合わせなどに用いることができる。

点が平面上に存在する平面スクリーンの場合と同様に、ドームスクリーンのような2次曲面上に点が存在する場合には以下の関係が成り立つ[4]。

スクリーンの2次曲面が次式で表されるとする。

$$X^T Q X = 0$$

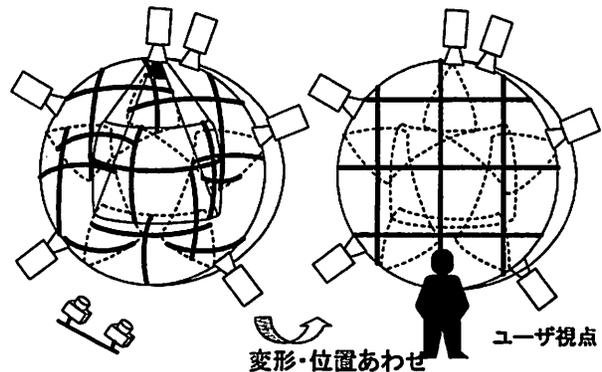
ここで Q は2次曲面式の係数を表す 4×4 行列、 X は3次射影座標である。この2次曲面上の点を任意の2つの視点から見た際に、それぞれの視点における点の射影座標 x' 、 x 間の関係は式(2)により表される。

$$x' = Hx - \left(q^T x \pm \sqrt{(q^T x)^2 - x^T Q_{33} x} \right) e \quad (2)$$

ここで、 H は平面射影変換、 e はエピポーラ、 Q_{33} 、 q は以下の式で表現される行列およびベクトルである。

$$Q = \begin{bmatrix} Q_{33} & q \\ q^T & 1 \end{bmatrix}$$

以上のように2つの視点間での関係が表されるため、プロジェクタからドームスクリーン上に投影した画像をカメラで読み込み、(2)式を求めることにより、カメラプロジェクタ間の画像変換を求めることができる。このようにしてすべてのカメラプロジェクタ間の関係を求めることにより、プロジェクタ相互の関係を導き、プロジェクタ投影画像の補正パラメタを自動的に求めている。



2台のカメラで個々のプロジェクタ投影パターンを認識

図1: 自動調整

2.2 GPUによるリアルタイム変形

汎用のプロジェクタは平面スクリーンに投影することを想定して製造されている。このため、3DCGのレンダリング画像をそのままドームスクリーン上に投影すると、直線はスクリーンに沿って曲がり、画像に歪みが生じてしまう。そこで歪みを補正するようにあらかじめ逆変形したレンダリング画像を生成して投影する必要がある(図2)。

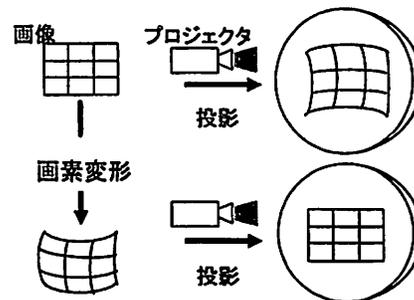


図2: リアルタイム変形

曲面への投影には、あらかじめ逆変形されたコンテンツを専用を用意するのが一般的である。静止画や動画のような固定された映像である場合、既存のコンテンツをあらかじめ逆変形しておけばよい。一方、インタラクティブにシーンビューを操作できる3DCGコンテンツでは、操作によってオブジェクトの見え方が変わるため、あらかじめオブジェクトを変形しておくことができず、シーンビューに応じてリアルタイムに変形をする必要がある。リアルタイムでの高速な変形画像生成は、プログラム可能なGPU(Graphics

Processing Unit)による演算で実現できる。

自動調整で得られたパラメタ e , Q_{33} , q をもとにした GPU 変形は、頂点シェーダを用いた 1 パスレンダリング方式[1]とピクセルシェーダを用いた 2 パスレンダリング方式[4]が提案されている。1 パスレンダリング方式では、頂点シェーダ内で 2 次曲面射影変換を行い、頂点の位置をずらすことで逆変形したジオメトリを作成して投影時の歪みを補正する。しかし、頂点のエッジは 2 次曲面射影変換後に 2 次曲線となるのが理想であり、この方式ではエッジ端点のみ歪みが補正されてエッジ中間点は 2 次曲面射影変換が適用されないため、直線のままとってしまう。端点間に頂点を追加するテセレーションを用いれば 2 次曲線に近似させていくことができるが、データ量そのものが多くなってしまふ。さらにこの場合、実行時のシーンビューやスクリーン形状によりスクリーン上での実投影長が異なるため、シーンに応じたテセレーションを行う必要性が生じる。また、頂点シェーダでは頂点属性が維持されないため、ライティングやフォグなどの頂点属性をすべてシェーダプログラムで再現しなくてはならず、任意のアプリケーションに対応できなくなる。

一方、2 パスレンダリング方式では、1 度シーンを通常通りにレンダリング(1 パス目)し、生成されたシーン画像をピクセルシェーダ内で 2 次曲面射影変換をしてレンダリング(2 パス目)することで、スクリーンのカーブに対応した歪曲画像を生成する。この方式ではレンダリング後のシーン画像を入力テクスチャとして変形するため、頂点属性が適用されている。変形はシーンによらず画素単位で行われ、直線のエッジも画素単位で変形される。よって 2 パスレンダリング方式は、歪み補正量を高精度に反映した高品質グラフィックス表示を実現できる。

式(2)では、カメラ視点からプロジェクタ間への射影変換を表しているが、カメラ視点 が被験者にとっての最適な視点となるわけではない。ドームスクリーンの特徴である広視角での没入感効果を得るためには、被験者はドームスクリーンの近くや内部にあることが望ましい。しかしながら、カメラ画像による調整ではドームスクリーン全体をカメラ視野に収める必要があり、ドームスクリーンが大きくなるほどカメラを離れた位置に置かなければならない。一般に多く用いられるカメラの画像から直接補正量を求める方式(ノンパラメトリック方式)では、カメラ位置と被験者の位置が一致している必要があるが、本方式では 2 次曲面投影変換をもとに補正量を計算する方式(パラメトリック方式)であるため、自動調整後も容易に被験者の視点(仮想視点)位置を変更できる。本方式では、カメラプロジェクタ間の射影変換、カメラ被験者間座標差分により、以下の方式で被験者プロジェクタ間の射影変換を求めることができる。

式(2)において表されるカメラからプロジェクタへの変換 Ψ_{CP} とカメラから仮想視点への変換 Ψ_{CV} により、仮想視点のピクセル座標 m_v は次式で表される。ここで、 m_p はプロジェクタのピクセル座標である。

$$m_v = \Psi_{CV} \Psi_{CP}^{-1}(m_p)$$

この m_v は仮想視点が変更されないかぎり変化しないため、シーンに応じて 2 次曲面射影変換を計算する必要はない。

3. 半球型ドームスクリーンシステム

3.1 システム概要

本システムは、以上で述べた 2 つの先行技術と独自の分散レンダリング技術により構成される。処理の流れは以下のようになる。

1. スクリーン、プロジェクタの位置関係を自動調整によって取得する(前処理)。
2. 3DCG アプリケーションを実行し、複数の PC に処理を割り当ててオフスクリーンレンダリングをする(リアルタイム処理)。
3. レンダリング画像を入力テクスチャとし、GPU で変形してレンダリングする(リアルタイム処理)。

図 3 に本システムの概要図を示す。本システムは操作 PC と、PC 間を接続する通信装置、複数の描画 PC から構成される。描画 PC はそれぞれプロジェクタに接続され、プロジェクタは投影領域がスクリーンをすべて覆うようにおおまかに配置される。詳細な位置合わせは自動調整により行い、レンダリングに必要なパラメタを取得する。操作 PC にてコンテンツを実行すると、分散レンダリングにより各描画 PC へ描画命令が渡り、適切な領域をレンダリングする。各 PC では、自動調整のパラメタをもとに GPU にて変形処理を行い、最終レンダリングをする。

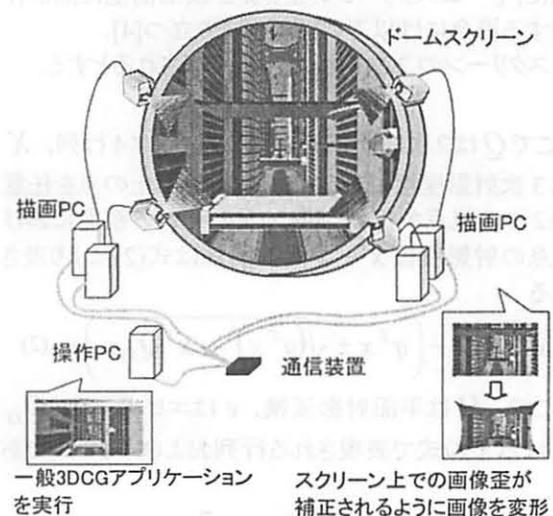


図 3: システム概要図

3.2 分散レンダリング

汎用プロジェクタによる高解像度のグラフィックスシステムを構築するために、プロジェクタを複数用いて映像を並び合わせることで、高解像度 3DCG を実現する。その際、従来では 3DCG コンテンツを投影領域に合わせて修正したり、スクリーンに合わせて作り込んだりする必要があった。また、あらかじめ 3DCG コンテンツを分散させて保持する方法もあるが、決められたシーンしか描画できず、マウスによってシーンビューを操作したり、イベントを発生させたりというインタラクティブな操作ができなくなってしまう問題があった。この対応策として、シーンビュー変更情報などを受信する API などをあらかじめ 3DCG アプリケーションに組み込む方法があるが、プログラムの修正など、高度な専門知識を要し、市販のアプリケーションには対応できない。

既存の 3DCG アプリケーションを修正なく実行するために、本システムでは、独自の分散レンダリング DLL (Dynamic Link Library) を構築し、OS が提供する 3DCG (OpenGL) ライブラリと置き換える。図 4 にソフトウェア構成図を示す。3DCG アプリケーションは、操作 PC 上にて実行する。3DCG とリンクできるように、分散レンダリング DLL は操作 PC 上に置かれ、各描画 PC 上には描画アプリケーションが置かれる。また、各描画アプリケーションには自動調整で得られたパラメタが設定されている。

OpenGL ライブラリの代わりに置かれた分散レンダリング DLL には、OpenGL ライブラリと同一のインタフェースを用意している。そのため、3DCG アプリケーションが OpenGL 関数を呼び出すと実際には分散レンダリング DLL が呼び出される。これにより、ユーザは 3DCG プログラムを変更することなく、また既存のバイナリ 3DCG アプリケーションもそのまま分散レンダリングを実行できる。分散レンダリング DLL は、実行時に 3DCG アプリケーションから発行されるビューやライト、テクスチャといった 3DCG 描画コマンドと描画データを次々に取得する。さらに取得した描画命令とデータを通信パケットに格納し、通信装置経由で各描画 PC へと送信する。ここで、分散レンダリング DLL はすべての描画 PC に対して同一の描画命令およびデータを送る。描画 PC 上では、描画アプリケーションが通信パケットを受け取り、3DCG の描画コマンドとデータに解釈してレンダリングする。このとき、命令を受け取ったすべての描画アプリケーションは同一シーンをオフスクリーンレンダリングするが、自動調整で得られたパラメタ e , Q_{33} , q が担当プロジェクタごとに異なっており、2 パス目で個々のプロジェクタ位置に応じた適切な画素変形が行われ、3DCG シーンが描画される。

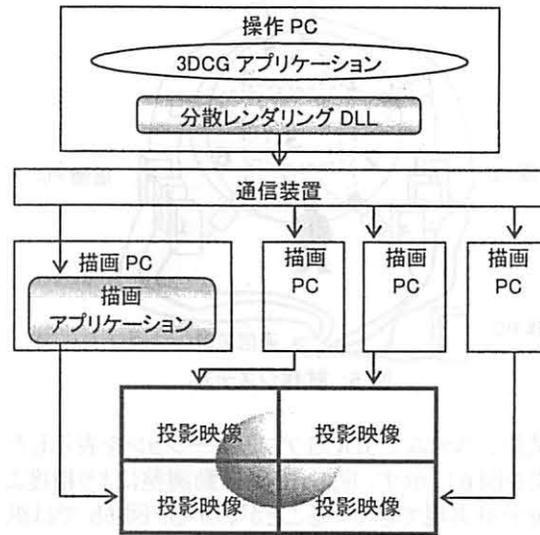


図 4: ソフトウェア構成

このように、レンダリング画像を配信するのではなく、描画コマンドを配信して複数の PC で分散的にレンダリングしているため、画像を拡大して表示する場合と違って解像度を生かした精細な描画が可能となる。より大きなスクリーンの場合にも、PC とプロジェクタの台数を増やせば、処理速度と精細さを維持したままいくらかでも投影領域を大きくできるスケーラブルなシステムである。

また、業界標準 3DCG ライブラリである OpenGL に対応しているため、Web ブラウザによる VRML (Virtual Reality Modeling Language) の表示や、CAD (Computer-Aided Design) アプリケーション、ゲームアプリケーションなども高解像度大画面上に表示できる。さらに、この分散レンダリング処理はすべてリアルタイムで行われるため、3DCG の特徴であるユーザアクションに対する応答や、動的な視界変更に対する制限もなく、インタラクティブなシーン操作が可能となっている。

4. 実験

4.1 試作システム

直径 3m のスクリーンに操作 PC 1 台と描画 PC 6 台 (PC: EPSON Endeavor, OS: WindowsXP Pro SP2, CPU: Pentium D 3.2GHz, MEM: 2.0GB RAM, GB: NVidia GeForce 7800 GTX) およびプロジェクタを 6 台用いて試作システムを構築した。描画 PC は通信装置を介して操作 PC に接続され、描画 PC に接続されたプロジェクタのうち 5 台は周辺部 (1,2,4,5,6) を投影し、残る 1 台は中央部 (3) を投影している (図 5)。

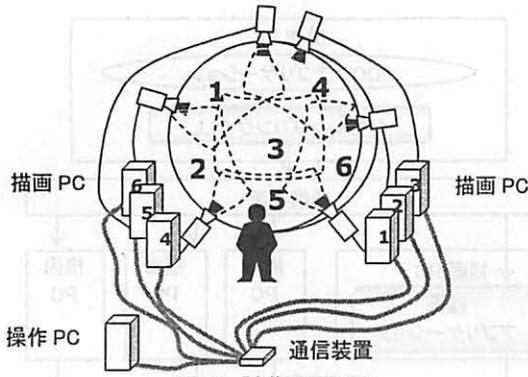


図 5: 試作システム

試作システムで 3DCG アプリケーションを表示した結果を図 6 に示す。図 6a では自動調整により精度よい表示が実現できていることがわかる。図 6b では汎用 CAD ソフトウェアによる表示、図 6c ではフライトシミュレータによる地形表示、図 6d ではアミューズメント向けの表示で未来都市をドライブする 3DCG アプリケーション(未来都市)を実行した。どのアプリケーションも単体 PC 用に作成されたものであるが、そのままドームスクリーン上で表示できている。

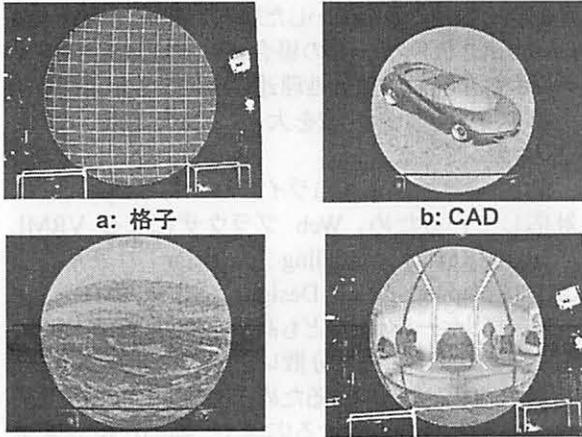


図 6: アプリケーション

図 7 に異なる視点からの様子を示す。仮想視点からは表示された建物がまっすぐ見えている(図 7a)。一方、仮想視点からずれると建物が大きく歪み、視点とスクリーン形状にあわせて変形表示がされていることがわかる(図 7b)。

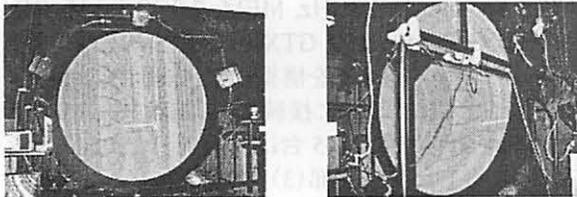


図 7: 異なる視点からの様子

4.2 性能評価

未来都市アプリケーションおよび SPEC Viewperf8.0 ensight-01 を用いて性能評価実験を行った。比較対象には通常の単体 PC 表示、および GPU による変形を行わないドーム表示を用いた。

測定結果を表 1 に示す。変形なしの場合と単体を比較すると、単体の約 45% の性能であることがわかる。これは分散処理による速度劣化である。さらに変形なしと本システムでは約 15% の差がみられるが、これは GPU の変形処理によるものである。

アプリケーション	未来都市	ensight-01
単体	73FPS	15.8FPS
変形なし	34FPS(46.6%)	6.7FPS(42.4%)
本システム	24FPS(32.9%)	4.4FPS(27.8%)

()内は単体性能比

表 1: 測定結果

5. まとめと今後の課題

2 次曲面射影変換による自動調整, GPU による画像変形を用い, 複数の PC による分散レンダリングを行うことで, 解像度の高い 3DCG を表示できるようにした。汎用 PC, 汎用プロジェクタ, 市販 3DCG アプリケーションを用いて構築できる汎用性に富んだシステムであり, より巨大なスクリーンでも高精細な表示が可能となるスケーラブルなシステムを実現した。また, 任意の OpenGL アプリケーションがそのままドームスクリーンに表示できるため, 既存のソフトウェア資産を有効に活用することができるようになった。

今後の課題として, レンダリング処理の最適化をはかり, システム全体の高速化を進めていく。また, 現状では動画を CG テクスチャとして表示するようなアプリケーションにより, 擬似的な動画表示もできるが, 動画を直接的に変形する機能を加えていきたい。

参考文献

- [1] R Raskar, J van Baar, T Willwacher, S Rao: "Quadric Image Transfer for Immersive Curved Screen Displays", Eurographics 2004
- [2] 柴野伸之, 澤田一哉, 竹林治雄: 「マルチプロジェクタを用いたスケーラブル大型ドームディスプレイ CyberDome の開発」, 日本バーチャルリアリティ学会論文誌, Vol.9 No.3 2004
- [3] 佐藤淳: 「コンピュータビジョン-視覚の幾何学-」, コロナ社
- [4] 緒方正人, 和田博之, 梶原景範, ジェロエンバンバー: 「プロジェクタ複合表示によるドームディスプレイシステム」, 情報処理学会第 120 回グラフィックスと CAD 研究会